



***Proyecto Fin de Carrera***

**DESARROLLO DE UNA HERRAMIENTA  
BUSINESS INTELLIGENCE PARA EL  
ANÁLISIS DE REDES SOCIALES  
ALMACENADAS EN GRAFOS**

**(Development of a Business Intelligence Tool for  
the Analysis of Social Networks Stored in Graph  
Databases)**

Para acceder al Título de

**INGENIERO EN INFORMÁTICA**

**Autora: Noelia Ruiz Martínez  
Directora: Marta Elena Zorrilla Pantaleón**

**Junio – 2013**

## **Agradecimientos**

Ahora que con este Proyecto pongo fin a estos años de carrera, me gustaría dedicarles unas líneas a todas las personas que, de un modo u otro, han estado presentes a lo largo de este tiempo.

En primer lugar, quisiera darle las gracias la directora de mi proyecto, la Dra. Marta Elena Zorrilla Pantaleón, por las horas dedicadas, por sus consejos y por su profesionalidad.

En segundo lugar, gracias a mis padres, porque sin ellos no habría llegado hasta aquí. Gracias por preocuparos, por aconsejarme, por animarme en los peores momentos y por hacerme las cosas mucho más fáciles.

En tercer lugar, a Mariano por intentar ayudarme siempre que lo he necesitado, por apoyarme todas esas veces que estuve a punto de tirar la toalla y, sobre todo, por soportarme cuando más difícil era.

En cuarto lugar, a las personas tan maravillosas que he conocido a lo largo de estos años de carrera. Ojalá podamos seguir riéndonos de tantas anécdotas durante muchos años más.

Por último, quiero acordarme de mis abuelos, porque sé lo orgullosos que estarían.

Muchas gracias a todos.

## Resumen

En los últimos años se ha producido un gran crecimiento de los servicios basados en redes sociales como, por ejemplo, Twitter o Facebook. Si se visualiza la disposición de los datos que conforman estas redes, se asemeja en gran medida a un grafo en el que los nodos representan usuarios y las aristas que los unen las relaciones que se establecen entre ellos. Es por ello que, generalmente, estas aplicaciones utilizan gestores orientados a grafos para el almacenamiento de la información, como es el caso de Twitter, que hace uso del gestor FlockDB; o para soportar parte de su funcionalidad, como ocurre en el caso de Facebook que integra aplicaciones basadas en grafos a través del protocolo Open Graph.

Esta importancia de las redes sociales en la actualidad se traduce también en la relevancia de la búsqueda de información interesante para el negocio a partir de los datos correspondientes a la actividad realizada por los usuarios y a las relaciones existentes entre ellos.

Por estos motivos, este Proyecto Fin de Carrera ha tenido como objetivo el desarrollo de una herramienta Business Intelligence que incluyera la definición y el cálculo de indicadores de interés para analizar redes sociales. Con este propósito se utilizaron conjuntos de datos reales de diferentes características y tamaños y se almacenaron en un gestor de bases de datos orientadas a grafos, siendo Neo4J el elegido. Se diseñó, además, un almacén de datos, o *data warehouse*, como soporte a la herramienta Business Intelligence ante la imposibilidad de realizar ciertas consultas directamente sobre las bases de datos en grafo y con objeto de que el sistema pudiera ser extendido fácilmente. Posteriormente, se construyó el correspondiente módulo de extracción, transformación y carga así como los informes a través de los cuales los usuarios finales podrán explotar la información.

**Palabras clave:** Redes sociales, Inteligencia de negocio, Bases de datos orientadas a grafos

## **Abstract**

In the last few years, services based on social networks such as Facebook or Twitter have experienced great growth. If data from these networks is visualized, it looks like a graph in which nodes represent users and edges the relationships between them. As a consequence, these applications generally use graph-oriented database management systems to store the information. This is the case of Twitter, which uses a graph-oriented management system called FlockDB; or like Facebook, which integrates applications based on graphs using the Open Graph protocol to provide part of its functionality.

Nowadays, this importance of social networks involves, also, the importance of looking for interesting information in business using data which corresponds to the activity of users and the relationships between them.

Therefore, the purpose of this Degree Project has been the development of a Business Intelligence tool which included the definition and calculation of different interesting indicators to analyze social networks. With that in mind, some real datasets with different features and sizes were used and stored using Neo4J, a graph-oriented database management system. A data warehouse was designed too as a support to the Business Intelligence tool facing the impossibility of answering some queries using only the graph database and in order to extend easily the tool. Afterwards, an extraction, transformation and load module was built just like the reports through which the end user will analyze the information.

**Keywords:** Social networks, Business Intelligence, Graph-oriented databases

# Índice

|                                                                  |      |
|------------------------------------------------------------------|------|
| Índice de ilustraciones.....                                     | viii |
| 1 Introducción.....                                              | 1    |
| 1.1 Motivación.....                                              | 1    |
| 1.2 Objetivo .....                                               | 2    |
| 1.3 Fases del proyecto.....                                      | 2    |
| 2 Bases de datos orientadas a grafos.....                        | 4    |
| 2.1 Introducción .....                                           | 4    |
| 2.2 Modelos de datos basados en grafos .....                     | 7    |
| 2.2.1 Modelos más representativos.....                           | 7    |
| 2.2.2 Características del modelo basado en grafos .....          | 9    |
| 2.2.3 Lenguajes de manipulación y consulta.....                  | 10   |
| 2.3 Sistemas gestores de bases de datos orientadas a grafos..... | 13   |
| 2.3.1 DEX .....                                                  | 13   |
| 2.3.2 HypergraphDB.....                                          | 14   |
| 2.3.3 OrientDB.....                                              | 14   |
| 2.3.4 InfiniteGraph .....                                        | 15   |
| 2.3.5 Neo4J.....                                                 | 16   |
| 2.3.6 Selección del gestor .....                                 | 17   |
| 3 Business Intelligence.....                                     | 18   |
| 3.1 Introducción .....                                           | 18   |
| 3.2 Fases de desarrollo de una solución BI .....                 | 19   |
| 3.2.1 Definición de requisitos .....                             | 19   |
| 3.2.2 Fuentes de información .....                               | 20   |
| 3.2.3 Proceso ETL .....                                          | 20   |
| 3.2.4 Almacén de datos o <i>data warehouse</i> .....             | 21   |
| 3.2.5 Explotación de los datos .....                             | 23   |
| 3.3 Herramientas comerciales BI .....                            | 24   |
| 4 Especificación funcional.....                                  | 27   |
| 4.1 Requisitos funcionales: métricas .....                       | 27   |
| 4.1.1 Métricas de alcanzabilidad .....                           | 28   |
| 4.1.2 Métricas de compromiso .....                               | 29   |
| 4.1.3 Métricas de influencia.....                                | 30   |
| 4.2 Conjuntos de datos.....                                      | 30   |
| 4.2.1 MovieLens.....                                             | 30   |
| 4.2.2 Flickr .....                                               | 31   |
| 4.2.3 DBLP.....                                                  | 33   |
| 5 Diseño de la solución.....                                     | 36   |
| 5.1 Bases de datos en grafo.....                                 | 36   |

|        |                                                                                                              |    |
|--------|--------------------------------------------------------------------------------------------------------------|----|
| 5.1.1  | Base de datos del conjunto de datos MovieLens .....                                                          | 36 |
| 5.1.2  | Base de datos del conjunto de datos Flickr .....                                                             | 37 |
| 5.1.3  | Base de datos del conjunto de datos DBLP .....                                                               | 38 |
| 5.2    | Almacén de datos.....                                                                                        | 40 |
| 5.2.1  | Tabla <i>USER_DIMENSION</i> .....                                                                            | 40 |
| 5.2.2  | Tabla <i>GROUP_USER</i> .....                                                                                | 40 |
| 5.2.3  | Tabla <i>USERSNAPSHOT</i> .....                                                                              | 40 |
| 5.2.4  | Tabla <i>USER_SESSION</i> .....                                                                              | 41 |
| 5.2.5  | Tabla <i>AGE</i> .....                                                                                       | 41 |
| 5.2.6  | Tabla <i>AREA</i> .....                                                                                      | 41 |
| 5.2.7  | Tabla <i>OCUPPATION</i> .....                                                                                | 41 |
| 5.2.8  | Tabla <i>PRODUCT_DIMENSION</i> .....                                                                         | 41 |
| 5.2.9  | Tabla <i>PRODUCTSNAPSHOT</i> .....                                                                           | 41 |
| 5.2.10 | Tabla <i>CLASSIFICATION</i> .....                                                                            | 41 |
| 5.2.11 | Tabla <i>GENRE</i> .....                                                                                     | 41 |
| 5.2.12 | Tabla <i>DATE_DIMENSION</i> .....                                                                            | 41 |
| 5.2.13 | Tabla <i>HOURL_DIMENSION</i> .....                                                                           | 42 |
| 5.2.14 | Tabla <i>FRIENDSHIP</i> .....                                                                                | 42 |
| 5.2.15 | Tabla <i>SCORE</i> .....                                                                                     | 42 |
| 5.2.16 | Tabla <i>RANGE</i> .....                                                                                     | 42 |
| 5.3    | Proceso ETL.....                                                                                             | 42 |
| 6      | Generación de informes .....                                                                                 | 48 |
| 6.1    | Herramienta Reporting Services.....                                                                          | 48 |
| 6.2    | Descripción de informes.....                                                                                 | 48 |
| 6.2.1  | Informe de crecimiento de usuarios.....                                                                      | 50 |
| 6.2.2  | Informe de crecimiento de usuarios activos .....                                                             | 51 |
| 6.2.3  | Informe de velocidad de propagación.....                                                                     | 51 |
| 6.2.4  | Informe de actividad en la red .....                                                                         | 52 |
| 6.2.5  | Informe de frecuencia de publicación.....                                                                    | 52 |
| 6.2.6  | Informe de frecuencia de voto .....                                                                          | 53 |
| 6.2.7  | Informe de frecuencia de nuevas amistades .....                                                              | 54 |
| 6.2.8  | Informe de proximidad .....                                                                                  | 54 |
| 6.2.9  | Informe de crecimiento de seguidores .....                                                                   | 55 |
| 6.2.10 | Informe del número de número de visitas, comentarios y votos recibidos por los productos de un usuario ..... | 56 |
| 6.2.11 | Informe del número de visitas, comentarios y votos recibidos por un producto.....                            | 57 |
| 6.2.12 | Informe de reciprocidad .....                                                                                | 58 |
| 6.2.13 | Informe de fortaleza de los enlaces.....                                                                     | 58 |

|        |                                                      |    |
|--------|------------------------------------------------------|----|
| 6.2.14 | Informe de fidelidad de amigos .....                 | 59 |
| 6.2.15 | Informe del número de votos por proximidad .....     | 60 |
| 6.2.16 | Listas TOP10 .....                                   | 61 |
| 6.3    | Seguridad .....                                      | 63 |
| 6.4    | Mejoras de rendimiento.....                          | 64 |
| 7      | Conclusiones y líneas futuras.....                   | 65 |
| 7.1    | Conclusiones.....                                    | 65 |
| 7.2    | Líneas futuras.....                                  | 66 |
|        | Bibliografía .....                                   | 67 |
|        | Anexo I. Captura del almacén de datos diseñado ..... | 70 |

## Índice de ilustraciones

|                         |                                                                                                                |    |
|-------------------------|----------------------------------------------------------------------------------------------------------------|----|
| <b>Ilustración 2.1</b>  | Representación de un grafo dirigido a la izquierda y de un grafo no dirigido a la derecha.....                 | 5  |
| <b>Ilustración 2.2</b>  | Evolución de los modelos basados en grafos.....                                                                | 7  |
| <b>Ilustración 2.3</b>  | Grafo diseñado siguiendo el modelo <i>Property Graph</i> .....                                                 | 9  |
| <b>Ilustración 2.4</b>  | Script para importar un grafo utilizando Gremlin.....                                                          | 11 |
| <b>Ilustración 2.5</b>  | Script para consultar sobre el grafo importado utilizando Gremlin .....                                        | 12 |
| <b>Ilustración 2.6</b>  | Script para recorrer un grafo utilizando Gremlin .....                                                         | 12 |
| <b>Ilustración 2.7</b>  | Script de una consulta realizada con el lenguaje SPARQL.....                                                   | 12 |
| <b>Ilustración 2.8</b>  | Consulta de OrientDB que obtiene los actores que han trabajado en películas cuyo productor es J.J. Abrams..... | 15 |
| <b>Ilustración 2.9</b>  | Consulta Cypher que retorna los amigos y los amigos de amigos de cada nodo.....                                | 16 |
| <b>Ilustración 3.1</b>  | Componentes de una solución BI.....                                                                            | 19 |
| <b>Ilustración 3.2</b>  | Esquema multidimensional en estrella .....                                                                     | 22 |
| <b>Ilustración 3.3</b>  | Cubo OLAP .....                                                                                                | 24 |
| <b>Ilustración 4.1</b>  | Grafo genérico.....                                                                                            | 27 |
| <b>Ilustración 5.1</b>  | Grafo del conjunto de datos MovieLens .....                                                                    | 36 |
| <b>Ilustración 5.2</b>  | Grafo del conjunto de datos Flickr.....                                                                        | 38 |
| <b>Ilustración 5.3</b>  | Grafo del conjunto de datos DBLP .....                                                                         | 39 |
| <b>Ilustración 5.4</b>  | <i>Data staging</i> para el conjunto de datos MovieLens.....                                                   | 43 |
| <b>Ilustración 5.5</b>  | <i>Data staging</i> para el conjunto de datos Flickr .....                                                     | 44 |
| <b>Ilustración 5.6</b>  | <i>Data staging</i> para el conjunto de datos DBLP.....                                                        | 44 |
| <b>Ilustración 5.7</b>  | Proceso de carga en el <i>data staging</i> de Flickr .....                                                     | 45 |
| <b>Ilustración 5.8</b>  | Proceso de carga de la tabla <i>FRIENDSHIP</i> .....                                                           | 46 |
| <b>Ilustración 5.9</b>  | Proceso de carga de los datos del <i>data staging</i> de Flickr en el almacén de datos.....                    | 47 |
| <b>Ilustración 5.10</b> | Eliminación de las votaciones de Flickr con datos erróneos de fecha y hora.....                                | 47 |
| <b>Ilustración 6.1</b>  | Identificación de usuario en la interfaz web .....                                                             | 49 |
| <b>Ilustración 6.2</b>  | Formulario de inicio de la interfaz .....                                                                      | 49 |
| <b>Ilustración 6.3</b>  | Carpeta de informes sobre Flickr.....                                                                          | 50 |
| <b>Ilustración 6.4</b>  | Carpeta de informes con métricas de alcance .....                                                              | 50 |
| <b>Ilustración 6.5</b>  | Informe de crecimiento de usuarios.....                                                                        | 51 |
| <b>Ilustración 6.6</b>  | Gráfica del crecimiento de usuarios.....                                                                       | 51 |
| <b>Ilustración 6.7</b>  | Informe de velocidad de propagación .....                                                                      | 52 |
| <b>Ilustración 6.8</b>  | Informe de actividad .....                                                                                     | 52 |
| <b>Ilustración 6.9</b>  | Informe de frecuencia de publicación.....                                                                      | 53 |
| <b>Ilustración 6.10</b> | Informe de frecuencia de voto.....                                                                             | 53 |
| <b>Ilustración 6.11</b> | Gráfica de porcentaje de voto por rango de edad .....                                                          | 54 |
| <b>Ilustración 6.12</b> | Informe de frecuencia de amistades .....                                                                       | 54 |
| <b>Ilustración 6.13</b> | Informe de proximidad en amistades .....                                                                       | 55 |
| <b>Ilustración 6.14</b> | Gráfica de proximidad en amistades .....                                                                       | 55 |
| <b>Ilustración 6.15</b> | Informe de proximidad en votos.....                                                                            | 55 |
| <b>Ilustración 6.16</b> | Informe de crecimiento de seguidores .....                                                                     | 56 |
| <b>Ilustración 6.17</b> | Informe del total de comentarios, visitas y votos recibidos por usuario .....                                  | 56 |
| <b>Ilustración 6.18</b> | Informe del máximo de comentarios, visitas y votos recibidos por usuario.....                                  | 57 |
| <b>Ilustración 6.19</b> | Informe del número de votos por producto .....                                                                 | 57 |
| <b>Ilustración 6.20</b> | Gráfica de la valoración media agrupada por sexo .....                                                         | 58 |
| <b>Ilustración 6.21</b> | Gráfica del número de votos agrupado por sexo .....                                                            | 58 |
| <b>Ilustración 6.22</b> | Informe de reciprocidad de amistades .....                                                                     | 59 |
| <b>Ilustración 6.23</b> | Informe de fortaleza de los enlaces.....                                                                       | 59 |



|                         |                                                                        |    |
|-------------------------|------------------------------------------------------------------------|----|
| <b>Ilustración 6.24</b> | Informe de fidelidad de amigos .....                                   | 60 |
| <b>Ilustración 6.25</b> | Informe de proximidad de voto .....                                    | 60 |
| <b>Ilustración 6.26</b> | Informe de los 10 productos más y mejor valorados .....                | 61 |
| <b>Ilustración 6.27</b> | Informe de los 10 usuarios con más seguidores.....                     | 61 |
| <b>Ilustración 6.28</b> | Informe de los 10 usuarios con más productos .....                     | 62 |
| <b>Ilustración 6.29</b> | Informe de los 10 días con mayor actividad.....                        | 62 |
| <b>Ilustración 6.30</b> | Informe de las 10 horas con mayor actividad.....                       | 63 |
| <b>Ilustración 6.31</b> | Interfaz web vista por un usuario.....                                 | 63 |
| <b>Ilustración 6.32</b> | Interfaz web vista por un administrador.....                           | 64 |
| <b>Ilustración 6.33</b> | Cubo diseñado para calcular el número de votos recibido por usuario .. | 64 |
| <b>Ilustración A.1</b>  | Diseño del almacén de datos .....                                      | 70 |

# 1 Introducción

Este primer capítulo describe el contexto en el que se enmarca este Proyecto Fin de Carrera exponiendo las necesidades de las organizaciones derivadas de su presencia en redes sociales. Asimismo, se enumeran sus objetivos y se describen las fases llevadas a cabo para su realización.

## 1.1 Motivación

A lo largo de los últimos años, y gracias a la aparición y al éxito de las redes sociales, se ha producido un gran auge del llamado *marketing viral*. Este marketing utiliza información procedente de medios como blogs, correos electrónicos y, más recientemente, redes sociales con objeto de mejorar la posición de las empresas en el mercado. Analizando la información transmitida en este “boca a boca” electrónico, las empresas y organizaciones pueden conocer información muy relevante y en tiempo real que les ayude en la toma de decisiones como, por ejemplo, la valoración de sus productos, las críticas recibidas, conocer quiénes son sus clientes fieles, el perfil de sus detractores y qué clientes son más influyentes así como compararse con sus competidores. Otra de las ventajas que supone este tipo marketing es su coste, muy bajo en comparación con el tradicional.

Uno de los motivos que ha impulsado el marketing viral es el éxito que han experimentado las redes sociales en los últimos años. Cada vez son más las empresas presentes en ellas a través de, por ejemplo, páginas en Facebook o perfiles en Twitter. Los datos extraídos de estas redes sociales (por ejemplo, comentarios de opinión o perfiles de los usuarios que siguen la página) unidos a datos transaccionales permiten a las empresas extraer información referente a las características de los clientes fieles y potenciales (datos demográficos como la edad, el sexo o el nivel de estudios), determinar qué alcance tienen determinados productos (qué valoración reciben o a qué velocidad se propaga información referente a ellos), conocer qué usuarios son más influyentes en una red social (por ejemplo, por cada seguidor de una página ver cuántos de sus amigos se han unido a la página posteriormente o cuántos comentarios reciben las publicaciones que realiza) o cuál es el grado de compromiso de los usuarios (si el número de seguidores crece o no, cuántas visitas realizan, con qué frecuencia publican contenido, etc.). Esta información resulta muy útil a las empresas para, por ejemplo, diseñar campañas de marketing, determinar el volumen de fabricación de un producto o comprobar si un producto ha cumplido las expectativas de los usuarios. Para este fin, lo más adecuado es la definición de una serie de métricas o indicadores que permitan visualizar la información mencionada previamente con objeto de utilizarla durante los procesos de toma de decisiones. Dado que la información procede de muy diversas fuentes (redes sociales, servicios transaccionales de la organización y de otras fuentes disponibles en la empresa), una solución basada en un almacén de datos o *data warehouse* será la más apropiada.

Se ha de decir que, a pesar de la difusión e impacto de las redes sociales en el mercado, aún no se dispone de una definición clara y concreta de indicadores que puedan ser extraídos de las redes sociales y utilizados por las empresas en el proceso de toma de decisiones, motivo por el que este proyecto tiene por objeto definir estos indicadores así como ofrecer una implementación viable.

## 1.2 Objetivo

El objetivo de este Proyecto Fin de Carrera es la construcción de una aplicación de inteligencia de negocio que permita a las empresas y organizaciones analizar la información procedente de las redes sociales en las que participan. Esta herramienta debe cumplir con las siguientes premisas:

- **Mostrar métricas orientadas a medir la alcanzabilidad, compromiso e influencia de un usuario en la red social.** Con las métricas de alcanzabilidad se quiere medir la propagación de comentarios sobre productos o contenidos y su valoración; con las de compromiso, el grado de participación e implicación de los usuarios; y con las de influencia, conocer cómo afecta la actividad realizada por un usuario en otros. Para aplicar estas métricas y demostrar que son suficientemente generales, se utilizarán conjuntos de datos procedentes de redes sociales reales con diferente organización y objetivos.
- **Ofrecer una solución de inteligencia de negocio genérica para redes sociales basadas en grafos.** Inicialmente, se estudiará su implementación sobre una base de datos orientada a grafos y, si no resulta factible, se realizará una solución basada en almacenes de datos.
- **Ofrecer una interfaz fácil de usar e informes sencillos de interpretar.** Con el fin de que los usuarios finales puedan acceder a las métricas definidas de una forma sencilla, se implementarán informes sencillos y accesibles que recojan numérica y gráficamente esta información.

## 1.3 Fases del proyecto

La realización del proyecto se organizó en las fases que se mencionan a continuación y con el consiguiente conjunto de tareas:

1. **Estudio y análisis del problema.** Esta fase estuvo orientada a investigar los trabajos desarrollados hasta la fecha en el campo del marketing viral así como conocer qué tipo de bases de datos dan soporte a las redes sociales y el tipo de información que habitualmente se almacena en ellas. En concreto, las tareas abordadas fueron las siguientes:
  - a. **Búsqueda bibliográfica:** obtención de información sobre bases de datos orientadas a grafos y sobre indicadores para extraer información de redes sociales.
  - b. **Análisis comparativo de gestores de bases de datos orientadas a grafos.**
  - c. **Búsqueda de conjuntos de datos procedentes de redes sociales:** para conseguir una mayor generalidad en la solución, los datos buscados debían ser de diversa índole y con ciertas características de forma que, además, pudieran ser aprovechados lo máximo posible a la hora de extraer información de ellos.
2. **Especificación de requisitos.** Esta fase tuvo como objetivo la definición de las métricas a las que la solución final debía dar respuesta. Para ello, se estudiaron diferentes artículos en los que se recogían los indicadores más utilizados en el ámbito de las redes sociales y el marketing viral y se propuso un conjunto de indicadores generales.

3. **Especificación de diseño y desarrollo.** A lo largo de esta fase se diseñó la arquitectura lógica y física de la solución Business Intelligence para su posterior implementación, para lo cual se llevaron a cabo diversas tareas:
  - a. **Diseño de las bases de datos en grafo de cada uno de los conjuntos de datos elegidos.**
  - b. **Diseño del almacén de datos:** en base a las métricas propuestas, se diseñó una base de datos multidimensional adecuada para darles respuesta.
  - c. **Procesos de carga de datos:** la finalidad de esta tarea fue la implementación de procesos ETL (extracción, transformación y carga) para el traslado de los datos almacenados en el grafo y de los cálculos realizados sobre él a la base de datos multidimensional diseñada en el punto anterior.
  - d. **Diseño de informes:** en esta tarea se diseñaron e implementaron una serie de informes con información tanto numérica como gráfica para la consulta y explotación de las métricas definidas.
4. **Pruebas y validación.** Por último se procedió a la fase de pruebas y validación en la que se verificó la exactitud de las métricas calculadas y se comprobaron aspectos referentes a la seguridad en el acceso de los diferentes roles de usuario.

## 2 Bases de datos orientadas a grafos

En este capítulo se introduce el concepto de base de datos orientada a grafos para, posteriormente, describir sus modelos más representativos, sus características fundamentales y los lenguajes de consulta existentes. Para finalizar se realiza una breve descripción de los gestores analizados para la realización de este proyecto.

### 2.1 Introducción

Las bases de datos relacionales llevan más de 30 años dominando el mercado de la gestión de datos transaccional. Sin embargo, el incremento de la cantidad de información a almacenar y la heterogeneidad de los datos a procesar, evidenciaron sus limitaciones. En concreto, fueron las redes sociales las que impulsaron un nuevo concepto de base de datos conocido como NoSQL o BigData y que se diferencia del modelo relacional en varios aspectos:

- No utiliza SQL como principal lenguaje de consulta.
- Proporciona estructuras diferentes a las que utiliza el modelo relacional (estructuras fijas de tipo tabla) para adaptarse mejor a las necesidades de cada problemática.
- No permite el uso de operaciones de tipo JOIN<sup>1</sup>.
- No garantiza que se cumplan las propiedades ACID (Atomicity, Consistency, Isolation and Durability)<sup>2</sup>, lo que puede llevar a pérdidas de información o inconsistencias pero se aseguran otras propiedades como la disponibilidad de los datos.

Con estas características se persigue dar una mayor libertad a la hora de diseñar bases de datos, puesto que no exige que se organice la información en tablas y, además, se evitan los cuellos de botella que pueden producirse si, por ejemplo, se aplican las propiedades ACID a un gran número de transacciones.

Dentro de estas bases de datos NoSQL destacan las orientadas a clave-valor<sup>3</sup>, las orientadas a columnas<sup>4</sup>, las orientadas a documentos<sup>5</sup> y las orientadas a grafos, siendo estas últimas las empleadas para la realización de este proyecto y de las que se hablará a lo largo de este capítulo.

A grandes rasgos, una base de datos orientada a grafos consiste en almacenar los datos en un grafo y realizar su manipulación y consulta mediante operaciones propias de ellos. En matemáticas, un grafo se define como un conjunto de objetos (llamados nodos o vértices) que pueden unirse mediante enlaces (conocidos como aristas o arcos), los cuales representan relaciones binarias entre nodos. Matemáticamente, se definiría así:

---

<sup>1</sup> Sentencia del lenguaje SQL que permite combinar registros de dos o más tablas pertenecientes a una base de datos relacional.

<sup>2</sup> Mediante las propiedades de atomicidad, consistencia, aislamiento y durabilidad se garantiza que una serie de operaciones sean consideradas como una transacción.

<sup>3</sup> Permite el almacenamiento sin seguir un esquema definido: cada dato se introduce por su clave haciendo que su valor pueda ser cualquier cadena de bits.

<sup>4</sup> Se asemeja al modelo relacional pero, a diferencia de este, almacena los datos por columnas en vez de por filas.

<sup>5</sup> No se almacenan datos, sino documentos identificados con una clave única.

$$G = (V, E)$$

Donde  $V$  representaría el conjunto de nodos y  $E$  el conjunto de aristas que relaciona estos nodos. Lo habitual es que  $V$  sea finito, siendo el orden del grafo el número de nodos que lo forman y representado mediante  $|V|$ . Asimismo, cada nodo consta de un grado, que indica el número de aristas que tiene. A su vez, un grafo puede ser dirigido o no dirigido. En el primer caso, las aristas tienen dirección de forma que se distinguirá entre nodos origen y nodos destino. En el segundo caso, las aristas relacionarán dos nodos sin asignar un nodo origen y un nodo destino. En la Ilustración 2.1 se muestra una imagen de ejemplo de cada tipo de nodo descrito.



**Ilustración 2.1** Representación de un grafo dirigido a la izquierda y de un grafo no dirigido a la derecha

Asimismo, los grafos disponen de propiedades como la adyacencia (si dos nodos están conectados mediante una arista se dice que son adyacentes) y los caminos, los cuales indican un recorrido entre nodos adyacentes. Estos últimos son de vital importancia en muchas aplicaciones, como por ejemplo en medios de transporte para estimar qué ruta es la más corta o en una red social para calcular cuántas relaciones de amistad separan a un usuario de otro. Estos caminos, a su vez, pueden tener diferentes costes o pesos que se calculan en base a la suma del coste asignado a cada arista. Debido a su trascendencia, existe una gran cantidad de algoritmos<sup>6</sup> para calcular el camino más corto y/o menos costoso existente entre dos nodos.

Por su naturaleza y las operaciones que se pueden realizar sobre ellos, el uso de grafos como estructura de datos supone una serie de ventajas, algunas de las cuales se detallan a continuación:

- **Modelo más natural.** En algunos ámbitos, un grafo muestra los datos de una forma más cercana a la realidad, a la vez que permite almacenar todos los datos de una entidad en el nodo que la representa. Además, la información referente a las relaciones de cada una de estas entidades puede visualizarse de forma sencilla puesto que aparecen representadas como aristas unidas al nodo correspondiente a la entidad.
- **Consultas referidas a la estructura.** Como se mencionó previamente, una de las principales características de los grafos son los caminos y, más concretamente, el cálculo del camino más corto y/o menos costoso. La posibilidad de realizar este tipo de consultas referentes a la estructura proporciona a los usuarios un alto nivel de abstracción puesto que no es necesario que conozcan la estructura de la base de datos hasta el más mínimo detalle para llevar a cabo operaciones de este tipo.

---

<sup>6</sup> Algunos de los algoritmos más importantes en el recorrido de grafos son la búsqueda en anchura (BFS), la búsqueda en profundidad (DFS), la búsqueda A\*, el algoritmo de Dijkstra o el algoritmo de Floyd-Warshall.

- **Implementación eficiente.** Gracias a su estructura, las implementaciones de estas bases de datos pueden ofrecer algoritmos basados en grafos haciendo que ciertas operaciones, que resultarían complejas en otros tipos de almacenamiento, sean eficientes y sencillas.

A la hora de implementar estas bases de datos, varios autores han definido diversas formas de llevarlo a cabo. Una de ellas, la descrita por Hidders y Paradaens (1993), limita el uso de los grafos a la representación de instancias estando los datos almacenados en bases de datos construidas sobre otros modelos. De forma similar las definen Graves, Bergeman y Lawrence (1995), pero definiendo la base de datos sobre la que trabaja el grafo como una base perteneciente al modelo relacional. Contrarias a estas definiciones son las dadas por Levene y Loizou (1995), para los cuales el grafo funcionaría como la estructura de datos subyacente y sus instancias se trabajarían mediante otros modelos; y por Amam y Scholl (1992), según los cuales el grafo sólo tendría la función de organizar los datos. Otra implementación es la descrita por Gyssens, Paredaens, Den Bussche y Gucht (1990), que ofrece la posibilidad de utilizar una base de datos orientada a objetos como si fuera orientada a grafos definiendo para ello objetos que representen aristas, nodos y grafos completos así como sus operaciones.

Para trabajar con cualquiera de estas opciones de implementación del modelo orientado a grafos serán necesarias una serie de operaciones de manipulación, creación y acceso que, como se indicó, estarán orientadas a grafos. Además, serán necesarias constantes de integridad, identidad y referencia para su correcto funcionamiento. Algunos de estos aspectos asemejan las bases de datos orientadas a grafos a otros tipos de bases de datos más extendidos, como el relacional, el semántico o el orientado a objetos, lo que puede llevar a pensar si realmente los grafos son de utilidad a la hora de resolver ciertos aspectos o simplemente son una solución más. Aunque estos modelos mencionados podrían ser capaces de resolver algunas problemáticas con mayor o menor esfuerzo, lo cierto es que el modelo de grafos es el que se adapta de forma más sencilla y natural a la implementación de aplicaciones en las que las interconexiones son un aspecto clave. Algunas de estas aplicaciones están muy presentes en la vida diaria, como por ejemplo las redes de intercambio de información, las redes telefónicas, las redes de transporte, las redes biológicas en el campo de la medicina y, más recientemente, las redes sociales.

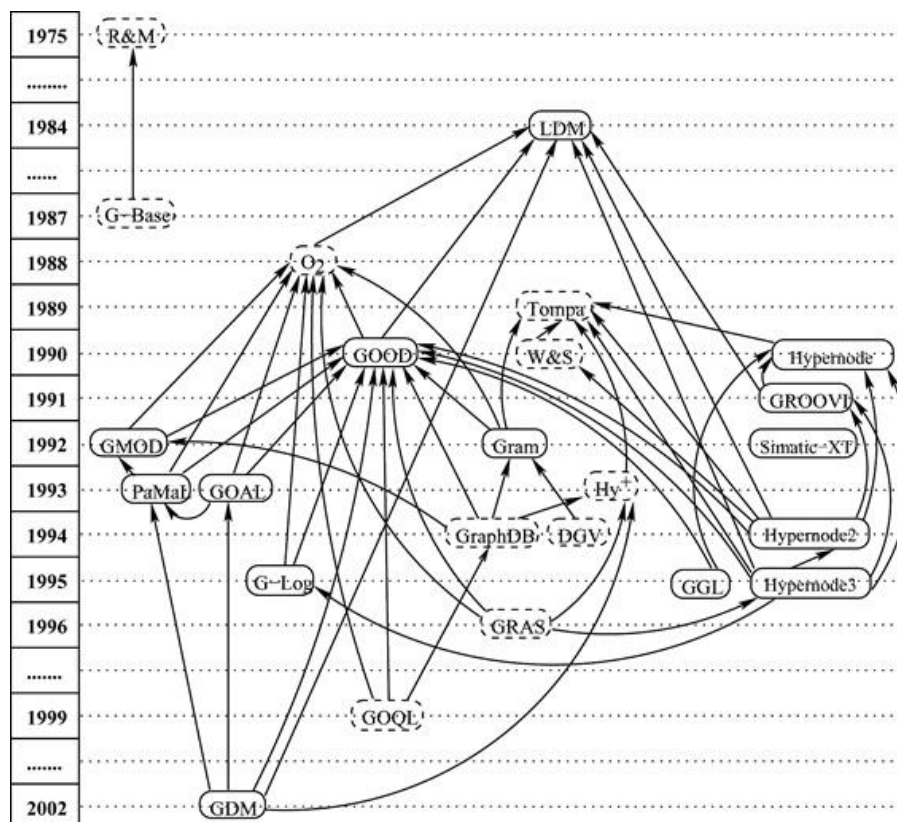
En estas últimas, un nodo representaría a una persona o a un grupo de personas y cada arista las relaciones entre ellos. Debido al gran éxito que comenzaron a experimentar alrededor de los años 2004 y 2005, se ha incrementado desde entonces la actividad en áreas referentes a su análisis, visualización y procesado haciendo así que las bases de datos basadas en grafos hayan adquirido una mayor relevancia. Sin ir más lejos, Mark Zuckerberg, creador de Facebook, acuñó el término “grafo social” (2007) para referirse a las relaciones existentes entre los usuarios de su red social. Asimismo, Facebook pone a disposición de los desarrolladores de aplicaciones una API para trabajar con la información de este grafo social<sup>7</sup>.

---

<sup>7</sup> Véase <https://developers.facebook.com/docs/getting-started/graphapi/>

## 2.2 Modelos de datos basados en grafos

A partir de los años 80, han sido varios los autores los que han definido modelos basados en grafos, alcanzando estas definiciones su mayor éxito a mediados de los 90. A partir de ese momento, los modelos de datos semi-estructurados los fueron desplazando debido al auge de la Web y el lenguaje XML. En la Ilustración 2.2 puede verse la evolución temporal de estos modelos así como las relaciones entre ellos.



**Ilustración 2.2** Evolución de los modelos basados en grafos

En los siguientes apartados se describen brevemente algunos de los modelos que aparecen en la Ilustración 2.2 así como las características que debe cumplir cualquier modelo basado en grafos y los lenguajes de consulta más utilizados en la actualidad.

### 2.2.1 Modelos más representativos

Como se observa en la Ilustración 2.2, Roussopoulos y Mylopoulos (1975) hicieron la primera aproximación al modelo orientado a grafos al proponer una red para almacenar datos en la que se tenía en cuenta la semántica de los datos, característica que hasta entonces había pasado inadvertida. No es hasta años más tarde cuando aparece el primer modelo basado en grafos, propuesto por Kuper y Vardi (1984), y que trataba de unificar mediante el uso de grafos los modelos relacionales, jerárquico y en red. La siguiente propuesta, realizada por Kunii (1987) y llamada G-Base, se basaba en la primera aproximación llevada a cabo en 1975 y su objetivo era la representación de datos con estructuras complejas.

Un par de años más tarde, se presenta el modelo O2 (Lécluse, Richard, & Vélez, 1988), orientado a objetos pero basado en una estructura de grafos. Similar a este modelo es el conocido como GOOD (Gyssens, Paredaens, Den Bussche, & Gucht, 1990), también



orientado a objetos y basado en grafos cuyo objetivo era utilizar grafos tanto para la manipulación como para la representación.

Este último modelo, GOOD, tuvo una gran influencia en los modelos que surgieron más tarde: GMOD (Andries, Gemis, Paredaens, Thyssens, & Den Bussche, 1992) propone nuevos conceptos de utilidad a la hora de representar las bases de datos orientadas a objetos en las interfaces de usuario; Gram (Amann & Scholl, 1992) está centrado en la representación de datos hiper-texto<sup>8</sup>; PaMal (Gemis & Paredaens, 1993) extiende GOOD permitiendo la representación de tuplas y conjuntos; GOAL (Hidders & Paradaens, 1993) complementa el modelo GOOD con nodos de asociación; GLog (Paredaens, Peelman, & Tanca, 1995) propone un lenguaje declarativo para trabajar con grafos; y finalmente se propone GDM (Hidders, 2002), que completa el modelo añadiendo la posibilidad de relaciones n-arias.

Además del modelo GOOD, destaca otra corriente cuyo objetivo era la generalización mediante el uso de grafos. La primera aproximación a este objetivo fue el modelo Hypernode (Levene & Poulouvassilis, 1990), que introdujo los grafos anidados o hiper-grafos, es decir, grafos en los que cada nodo (también llamados en este modelo hiper-nodos) puede contener, a su vez, grafos. Esta misma idea se utilizó para representar redes multi-escalares (Mainguenaud, 1992) y datos genéticos (Graves, Bergeman, & Lawrence, 1995). Otro modelo importante que también se basa en Hypernode es GROOVY (Levene & Poulouvassilis, 1991), que utiliza un modelo orientado a objetos pero se formaliza mediante el uso de nodos anidados. Este concepto de grafos anidados también se utiliza en otros muchos modelos, tanto en la consulta y visualización de los datos como en el acceso a ellos.

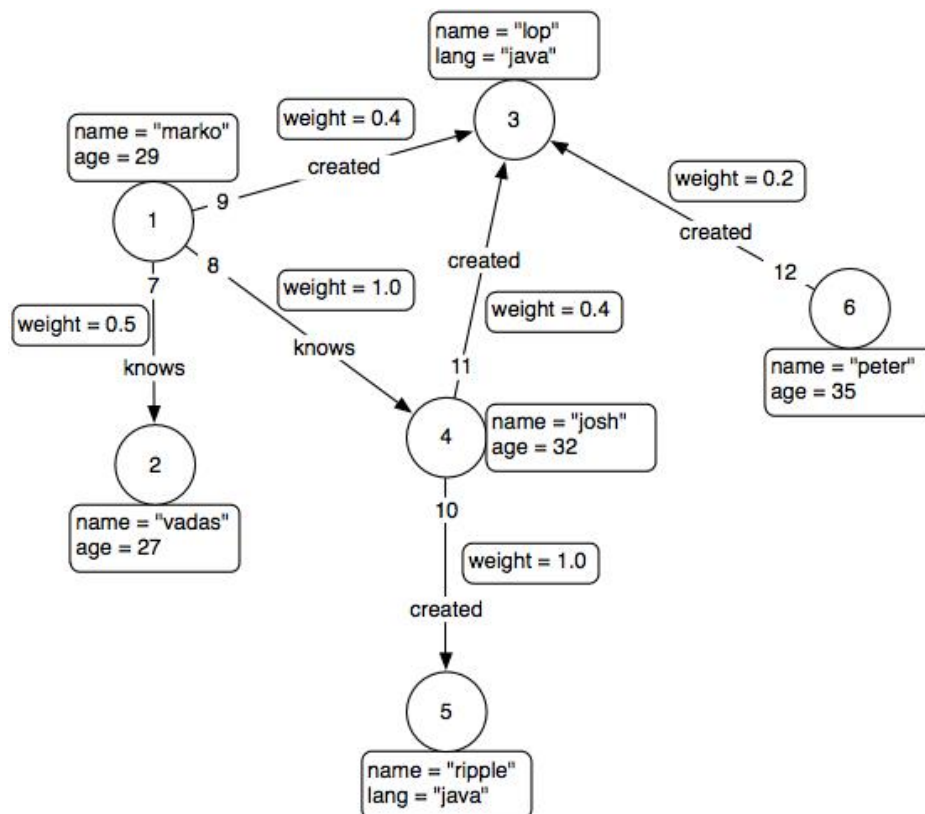
Además de los modelos descritos en los párrafos anteriores, existen otros motivados por problemas concretos, como GraphDB (Güting, 1994), que surgió gracias las redes de transporte y que está basado en las bases de datos orientadas a objetos pero con el objetivo de modelar y consultar grafos. Otro ejemplo es Graph Views (Gutiérrez, Pucheral, Thévenin, & J.M., 1994), ideado con el fin de añadir una capa de abstracción a la hora de trabajar con grafos almacenados en sistemas orientados a objetos. Finalmente, el modelo OEM (Papakonstantinou, García-Molina, & Andwidom, 1995), utiliza los grafos para emplearlos en el intercambio de fuentes de información muy heterogéneas entre sí.

Se ha de tener en cuenta que muchos de los modelos arriba mencionados no disponen de implementación real y, en caso de disponer de ella, su uso no está extendido. Sin embargo, aunque algunas de las implementaciones más utilizadas en la actualidad no se ajustan en su totalidad a los modelos descritos, sí que toman características de varios de ellos. Este es el caso del modelo conocido como Property Graph, uno de los más utilizados en implementaciones reales. Sus tres características principales son: (i) el uso de parejas clave-valor, (ii) ser dirigido y (iii) ser multi-relacional. La primera de sus características indica que los nodos y aristas que lo forman disponen de una serie de propiedades a las que se accede a través de pares clave-valor; la segunda, que las aristas pueden tener dirección (es decir, un nodo origen y un nodo de destino); y la tercera hace referencia a las relaciones que pueden darse entre dos nodos, pudiendo ser éstas de diferentes tipos y

---

<sup>8</sup> Almacenamiento de información utilizando una red de nodos conectados por enlaces. Cada nodo contiene información de tipo texto.

permitiendo que entre esos dos nodos existan relaciones de más de un tipo. Un ejemplo de grafo diseñado siguiendo este modelo es el que se muestra en la Ilustración 2.3.



**Ilustración 2.3** Grafo diseñado siguiendo el modelo *Property Graph*

Todos los grafos de este modelo tienen un conjunto de vértices y un conjunto de aristas. Cada uno de los vértices se caracteriza por disponer de un identificador único (en la imagen, el número que aparece dentro de cada nodo); un conjunto de aristas (tanto las entrantes como las salientes); y una colección de propiedades definidas por mapas clave-valor (en el grafo de la imagen, las propiedades serían *name*, *age* y *lang*). En el caso del conjunto de las aristas, cada una debe disponer también de un identificador único (indicado en el ejemplo como el número que aparece sobre cada arista); un vértice origen y un vértice destino; una etiqueta indicando de qué tipo es cada arista (en el caso del grafo de ejemplo los tipos serían *created* y *knows*); y una colección de propiedades que, al igual que en los nodos, son mapas clave-valor (en el ejemplo, todas las aristas tienen la misma propiedad, *weight*, independientemente del tipo al que pertenecen).

### 2.2.2 Características del modelo basado en grafos

A grandes rasgos, los modelos de datos orientados a grafos deberán ser capaces de representar entidades (unidades que existen por sí mismas) y las relaciones que pueden darse entre dichas entidades. Además de estas características fundamentales, también deberá tener otras, propias de otros modelos, como la presencia de atributos (modelo relacional), abstracciones (modelo semántico), objetos complejos (modelo orientado a objetos) y relaciones compuestas (modelo semi-estructurado). En función de las necesidades, se determinará si la mejor opción es trabajar con un grafo etiquetado, no etiquetado, dirigido, no dirigido o que utilice hiper-nodos. Independientemente de sus

características, existen ciertos aspectos que deberá cumplir todo modelo: cómo representar las entidades, cómo representar las relaciones y cómo definir constantes de integridad.

En primer lugar, las entidades deberán representarse tanto en el esquema del modelo como en sus instancias. Por su parte, el esquema deberá definir, en caso de que sea necesario, qué tipos de entidades pueden darse en el grafo, mientras que las instancias representarán tuplas o elementos concretos pertenecientes a los tipos definidos en el esquema. En función del modelo que se siga, deberán definirse otros conceptos como los hiper-nodos.

En segundo lugar, será necesaria la representación de las relaciones. Éstas pueden ser simples (unión de dos nodos pudiendo tener atributos o no) o complejas (por ejemplo, de tipo jerárquico y con semántica adicional, como operaciones de composición). En función del dato que representen, estas aristas podrán clasificarse de diferentes maneras. Si, por ejemplo, representan una propiedad de una entidad pueden considerarse atributos; si se limitan a representar una relación entre dos entidades, se dice que son relaciones de vecindad; si la relación puede considerarse como una entidad diferente a las que relaciona, se trata a dicha relación como una entidad. También puede darse el caso de que representen abstracciones o relaciones de herencia.

Finalmente, otra característica que es necesario definir aunque tenga menor importancia que las dos anteriores son las constantes de integridad. Estas constantes son reglas que definen una serie de estados de la base de datos de forma que esta sea consistente. No son propias de las bases de datos orientadas a grafos, sino que se han definido en otros modelos como el relacional, el orientado a objetos y el semi-estructurado. En el caso del modelo orientado a grafos, las constantes de integridad pueden hacer referencia a la consistencia de las instancias del esquema (es decir, exigir que las instancias se ajusten a las etiquetas, atributos o relaciones definidos en el esquema), a constantes de identidad (definidas mediante atributos o identificadores) y referenciales (identifican los nodos que unen una relación de forma similar a la *foreign-key*<sup>9</sup> del modelo relacional) y a constantes funcionales (dependencias que se dan cuando el valor de un atributo condiciona los valores de otros).

### 2.2.3 Lenguajes de manipulación y consulta

Siguiendo la definición de Codd (1980), un lenguaje de consulta es un conjunto de operadores o de reglas de inferencia que pueden aplicarse a las instancias de un esquema con el objetivo de manipular y consultar datos siguiendo la estructura del esquema y con posibilidad de diferentes combinaciones. En el modelo orientado a grafos, se han presentado diversos lenguajes para su consulta y la representación de resultados.

En el apartado 2.2.1 se mencionó la existencia de lenguajes de consulta creados en base a algunos modelos. Sin embargo, y al igual que ocurre con los modelos sobre los que se construyen, la mayoría de ellos o no están implementados o, en caso de estarlo, su uso no está muy extendido, por lo que en las líneas siguientes se describirán los lenguajes de consulta más extendidos con implementación real.

---

<sup>9</sup> Clave formada por una o varias columnas de una tabla que hace referencia a una o varias columnas de otra tabla.

Uno de los lenguajes más empleados actualmente es Gremlin, considerado un lenguaje específico de dominio e incluido en la interfaz Blueprints. Permite la consulta, el análisis y la manipulación de cualquier grafo que se haya implementado siguiendo la interfaz mencionada, la cual se basa en la definición del modelo Property Graph descrito anteriormente, siendo la más extendida en la mayoría de gestores actuales. Esta interfaz, similar a JDBC<sup>10</sup> pero conectando a una base de datos construida empleando grafos en lugar de a una base de datos relacional, permite realizar consultas utilizando el lenguaje Gremlin sobre grafos que siguen el modelo PropertyGraph. Además de un lenguaje de consulta, proporciona una serie de interfaces con diferentes propósitos, como la transformación de objetos, el uso de algoritmos basados en grafos o la implementación de un servidor para trabajar con estos grafos.

El lenguaje de consulta ofrecido por Blueprints, Gremlin, está construido con Groovy, considerado como un superconjunto de Java (por ello, todo lo que funciona en Java también funciona en Groovy) y que facilita la metaprogramación para crear lenguajes específicos de dominio. Puesto que Gremlin es un lenguaje de dominio construido sobre Groovy, no existirán problemas de compatibilidad con Java de forma que se podrán utilizar algoritmos escritos en Java dentro de consultas realizadas con Gremlin.

Al tratarse de un lenguaje específico de dominio, facilita operaciones que en otros lenguajes resultarían más complicadas. Por ejemplo, a la hora de llevar a cabo consultas complejas, Gremlin permitirá escribir en pocas líneas de código consultas que necesitarían de muchas más en otros lenguajes. Además de la simplicidad, garantiza la integridad de los datos tras operaciones como la actualización de nodos y la creación y eliminación de aristas. Para mostrar la simplicidad de Gremlin, a continuación se muestran una serie de consultas realizadas con este lenguaje a modo de ejemplo.

Para la realización de estos ejemplos, fue necesario importar el grafo mostrado en la Ilustración 2.3 y asignárselo a la variable de nombre `g`, lo cual queda recogido en el primer comando de la Ilustración 2.4. Como resultado de esta operación se mostrarán el número de nodos y aristas que forman el grafo. A continuación, y con el fin de trabajar con él en los siguientes ejemplos, se elegirá el nodo de identificador 1 y se le asignará la variable `v`, operación correspondiente al segundo comando de la Ilustración 2.4.

```
gremlin> g = TinkerGraphFactory.createTinkerGraph()  
==>tinkergraph[vertices:6 edges:6]  
gremlin> v = g.v(1)  
==>v[1]
```

**Ilustración 2.4** Script para importar un grafo utilizando Gremlin

En la Ilustración 2.5, y mediante la instrucción `v.outE`, se obtienen todas las aristas salientes del nodo designado por `v` acompañadas de su identificador y de los nodos que unen junto con el tipo al que pertenecen.

---

<sup>10</sup> JDBC (Java Database Connectivity) es una API para la ejecución de operaciones sobre bases de datos desde el lenguaje de programación Java.

```
gremlin> v.outE
==>e[7][1-knows->2]
==>e[9][1-created->3]
==>e[8][1-knows->4]
```

**Ilustración 2.5** Script para consultar sobre el grafo importado utilizando Gremlin

Finalmente, en la Ilustración 2.6 se muestra un ejemplo de recorrido. En este caso se obtendrán, partiendo del nodo 1, los nodos que estén a distancia 2 siguiendo las aristas salientes.

```
gremlin> v.out.out
==>v[5]
==>v[3]
```

**Ilustración 2.6** Script para recorrer un grafo utilizando Gremlin

Aunque en los ejemplos no se han llevado a cabo operaciones complejas, sí puede apreciarse la simplicidad de la que se hablaba al describir este lenguaje ya que, utilizando lenguajes de propósito general en lugar de propósito específico como en este caso, las instrucciones para ejecutar operaciones tan sencillas como las mostradas habrían sido más arduas.

Si bien el uso de Gremlin está muy extendido debido a la gran cantidad de gestores que siguen el modelo propuesto por Blueprints, existen otros lenguajes de consulta sobre grafos. Uno de ellos es SPARQL, ideado para realizar consultas sobre RDF (Resource Definition Framework), un estándar diseñado para la representación de formato de datos de la Web mediante grafos etiquetados y dirigidos. Por este motivo, SPARQL puede utilizarse para realizar consultas sobre bases de datos almacenadas en grafos siempre que el formato de dicho grafo sea el mismo que utiliza RDF. Además de la realización de consultas, también permite operaciones de creación, actualización y eliminación de datos.

Una de sus principales características es su sintaxis: su similitud con el lenguaje relacional SQL hace que los usuarios habituados a este lenguaje no tengan grandes problemas a la hora de utilizar SPARQL. En la Ilustración 2.7 se muestra una consulta sencilla en la que se puede comprobar esta similitud.

```
SELECT ?x WHERE {   tg:1 tg:knows ?x   }
```

**Ilustración 2.7** Script de una consulta realizada con el lenguaje SPARQL

El objetivo de esta consulta es obtener todos los nodos relacionados con el nodo de identificador 1 mediante una relación del tipo `knows`. El resultado serán los identificadores de todos los nodos relacionados con el nodo 1 mediante este tipo de relación y que en la consulta aparecen representados mediante la variable `?x`.

Aunque Gremlin y SPARQL son los principales lenguajes empleados, algunos gestores proporcionan sus propios lenguajes de consulta a la vez que permiten el uso de estos dos lenguajes. Un ejemplo de estos gestores son Neo4J y OrientDB, gestores que se

estudiarán en el siguiente apartado y que proponen, respectivamente, el lenguaje Cypher y una adaptación de SQL para grafos.

### 2.3 Sistemas gestores de bases de datos orientadas a grafos

A grandes rasgos, un Sistema Gestor de Bases de Datos (SGBD) es una aplicación que proporciona herramientas para el almacenamiento, modificación y extracción de la información almacenada en una base de datos. Además, suele ofrecer métodos para garantizar la integridad de los datos, controlar el acceso de usuarios y recuperar datos en caso de errores graves.

Existen diferentes SGBDs en función del tipo de base de datos con el que trabajen, las características de la máquina sobre la que funcionan o las necesidades de los usuarios. A la hora de elegir con qué gestor orientado a grafos se iba a realizar este proyecto se tuvieron en cuenta tres aspectos haciendo hincapié en los detalles siguientes:

- Además de trabajar sobre bases de datos orientadas a grafos, estos grafos debían cumplir ciertas características. Entre estas características destacaba el hecho de que permitieran atributos en los nodos y en las aristas. La mayoría de los gestores que se estudiaron permitían atributos en los nodos pero, sin embargo, en muchos de ellos no ocurría lo mismo con las aristas. Gestores muy conocidos como AllegroGraph<sup>11</sup>, VertexDB<sup>12</sup> o BigData<sup>13</sup> no cumplían esta característica, por lo que fueron descartados.
- Con los gestores que cumplían estas características mencionadas en el punto anterior, se examinaron aquellos que permitían el uso de lenguajes de programación conocidos y multiplataforma, como Java y C++.
- La naturaleza *open-source* de los gestores fue otro factor a la hora de decidir: se eligieron aquellos que fueran gratuitos o, al menos, dispusieran de versiones gratuitas que cubrieran las acciones a realizar.

Teniendo en cuenta estas características, se examinaron las diferentes opciones disponibles y se eligieron aquellas que las cumplían para, posteriormente, estudiarlas y compararlas con el fin de elegir la mejor opción. En los siguientes apartados se describen los gestores estudiados y se analizan sus pros y contras tras su uso.

#### 2.3.1 DEX

Desarrollado a partir del año 2006 por el grupo DAMA-UPC (Data Management de la Universidad Politécnica de Cataluña), DEX<sup>14</sup> es un sistema gestor de bases de datos orientadas a grafos programado sobre el lenguaje C++. Las bases de datos con las que trabaja quedan almacenadas en disco y hace uso de grafos dirigidos, etiquetados y con atributos tanto en nodos como en aristas, es decir, sigue el modelo Property Graph visto en el apartado 2.2.1 y trabaja sobre la interfaz Blueprints. Si bien al basarse en grafos se engloba dentro del paradigma NoSQL, soporta parcialmente las transacciones ACID aunque este paradigma no garantice que se cumplan.

---

<sup>11</sup> Véase <http://www.franz.com/agraph/allegrograph>

<sup>12</sup> Véase <http://data.story.lu/tag/vertexdb>

<sup>13</sup> Véase <http://www.systap.com/bigdata.htm>

<sup>14</sup> Véase <http://www.sparsity-technologies.com/dex>

Uno de los motivos por los que las bases de datos NoSQL se alejan de este tipo de transacciones es por los problemas que suponen en el rendimiento al trabajar con grandes cantidades de datos. Sin embargo, una de las principales características de este gestor es el rendimiento que, si bien no se obtiene sustituyendo las propiedades ACID, se consigue por medio de otras técnicas, como la división del grafo en varias partes independientes entre sí reduciendo de esta forma la operaciones de entrada/salida en disco; y el almacenamiento de los nodos y las aristas de forma similar a *bitmaps*<sup>15</sup>, lo que permite técnicas de compresión más eficientes.

Con estas medidas para la mejora del rendimiento, DEX permite el trabajo con bases de datos que, además de ser de gran tamaño, soporten una gran concurrencia de usuarios puesto que consigue tiempos de respuesta muy rápidos.

Aunque está programado sobre el lenguaje C++, ofrece su API, además de en C++, en Java y .NET, haciendo de él un gestor multiplataforma. Por ello, a la hora de comenzar a trabajar con DEX, se deberá descargar la librería correspondiente al lenguaje que se utilizará y se indicará, además, cuál será el tamaño de la base de datos puesto que se ofrecen varias posibilidades de descarga: desde el tamaño máximo soportado por la versión gratuita (un millón de objetos) hasta los tamaños correspondientes a las versiones de pago, que llegan a alcanzar los diez millones.

### 2.3.2 HypergraphDB

Aunque HypergraphDB<sup>16</sup> tiene como principal objetivo las bases de datos orientadas a grafos, también permite trabajar con bases de datos semánticas, relacionales y orientadas a objetos, lo que lo define como un gestor muy flexible.

Respecto al tipo de grafos con los que trabaja, utiliza el modelo Hypernode del que se habló en el apartado 2.2.1, pero con ciertas modificaciones de forma que una arista, además de poder apuntar a varios nodos, también pueda apuntar a varias aristas.

Su implementación está basada en Java, haciendo de ella una herramienta multiplataforma y se contempla la posibilidad de hacerlo también en C++. Los requisitos necesarios antes de trabajar con este gestor son, nuevamente, la importación de su librería en el proyecto Java en el que se vaya a trabajar con la base de datos.

### 2.3.3 OrientDB

Lanzado en el año 2010 y escrito en Java (considerado, por tanto, multiplataforma), OrientDB<sup>17</sup> es un gestor de base de datos que, aunque orientado a la gestión de documentos, soporta trabajar con bases de datos orientadas a grafos ya que las relaciones entre documentos se manejan como si fueran grafos. Además, permite otorgar diferentes grados de flexibilidad al esquema y crear roles de usuario para garantizar la seguridad.

A pesar de pertenecer al grupo de las bases de datos que siguen el paradigma NoSQL, cumple las propiedades ACID a la hora de realizar transacciones y además permite el uso del lenguaje relacional SQL con extensiones de forma que se puedan manejar relaciones sin hacer uso de JOIN, siendo este hecho una de las principales características

<sup>15</sup> Formato de imágenes. Véase [http://es.wikipedia.org/wiki/Imagen\\_de\\_mapa\\_de\\_bits](http://es.wikipedia.org/wiki/Imagen_de_mapa_de_bits)

<sup>16</sup> Véase <http://www.hypergraphdb.org/index>

<sup>17</sup> Véase <http://www.orientdb.org/>

que lo diferencia (en la Ilustración 2.8 se muestra un pequeño ejemplo de este lenguaje). Para conseguir solventar la pérdida de rendimiento que supone el uso de ACID, utiliza algoritmos de indexado propios de bases de datos documentales, como MVRB-Tree, combinación del árbol B+<sup>18</sup> y Red-Black<sup>19</sup>, con los que se consiguen inserciones y búsquedas rápidas.

La estructura de grafos que emplea sigue el modelo Property Graph y, además, está construido sobre la interfaz Blueprints, lo que permite el uso de todas las características de ésta además de su lenguaje, Gremlin.

```
select from (
  traverse Movie.actors, Actor.movies from (
    select from Movie where producer = "J.J. Abrams"
  ) while $depth <= 3
) where @class = 'Movie'
```

**Ilustración 2.8** Consulta de OrientDB que obtiene los actores que han trabajado en películas cuyo productor es J.J. Abrams

A la hora de trabajar con OrientDB es necesaria la importación de su librería Java y la puesta en funcionamiento de un servidor con licencia Apache, totalmente gratuito. Otra opción para utilizar este gestor es a través de línea de comandos.

### 2.3.4 InfiniteGraph

InfiniteGraph es un gestor de bases de datos orientadas a grafos con núcleo programado en C++ pero implementado en Java, lo que hace de él un gestor multiplataforma. Además, sigue el modelo de grafos Property Graph y utiliza la interfaz Blueprints.

A la hora de almacenar la información, ofrece dos posibilidades: hacerlo en la misma máquina en la que se trabaje o distribuir el almacenaje en varias máquinas, consiguiendo así ventajas en el rendimiento en caso de que se trabaje con muchos datos.

Además de la posibilidad de distribuir el almacenamiento, se ofrecen dos opciones a la hora de realizar transacciones para mejorar el rendimiento en operaciones de modificación. La primera de ellas, indicada si no se tienen muchos datos o es fundamental que se controle la consistencia de estos, exige que se cumplan todas las propiedades ACID en cada transacción. La segunda opción, recomendada cuando existen muchos datos, garantiza la consistencia pero sin exigir que se cumplan todas las propiedades ACID, consiguiendo de esta forma un menor impacto en el rendimiento. Otro mecanismo que utiliza a la hora de mejorar el rendimiento es almacenar físicamente los elementos a los que se acceda frecuentemente de forma contigua, mejorando así la navegación en las consultas.

Antes de comenzar a trabajar con InfiniteGraph es necesario descargar su librería Java para incluirla en el proyecto que albergará la solución. A la hora de realizar consultas,

---

<sup>18</sup> Tipo de estructura de datos en forma de árbol utilizada en la representación de colecciones ordenadas de objetos y que permite inserciones y borrados eficientes. Muy utilizados para la indexación.

<sup>19</sup> Tipo de árbol binario de búsqueda utilizado para organizar información compuesta por datos comparables.



ofrece la posibilidad de llevarlas a cabo mediante lenguaje de lógica de predicados o, por estar implementado sobre Blueprints, mediante Gremlin.

### 2.3.5 Neo4J

Implementado en Java y basado en grafos, Neo4J<sup>20</sup> nació en el 2007 y hoy en día es uno de los gestores de bases de datos orientadas a grafos más extendidos, siendo utilizado por empresas tan conocidas como Adobe, Cisco, Deutsch Telekom o Infojobs.

Descrito por sus desarrolladores como “*robusto, escalable y de alto rendimiento*”, emplea diferentes técnicas para lograr estos objetivos. En primer lugar, hace uso de transacciones ACID aunque pertenezca al paradigma NoSQL puesto que sus desarrolladores las consideran los cimientos de la fiabilidad de los datos. Por su implementación, los grafos pueden crecer almacenando mayores cantidades de datos sin que apenas se perciba en el rendimiento, siendo los recursos hardware la única barrera. Para solventar este problema, se ofrece la posibilidad de distribuir la base de datos en varios servidores.

Respecto a la estructura de grafos empleada, sigue el modelo Property Graph, por lo que hace uso de la interfaz Blueprints aprovechando así sus características, como el lenguaje de consulta Gremlin. Además de este lenguaje también puede utilizarse uno propio del gestor, Cypher, del que se muestra un ejemplo en la Ilustración 2.9, caracterizado por acercarse al lenguaje natural más que Gremlin.

```
START me=node(*) MATCH me-->friend-[?]->friend_of_friend
RETURN friend, friend_of_friend
```

**Ilustración 2.9** Consulta Cypher que retorna los amigos y los amigos de amigos de cada nodo

A la hora de trabajar con Neo4J, existe la posibilidad de integrarlo en Java, para lo que se deberán descargar sus librerías; o instalando su servidor, de forma que se pueda trabajar con él mediante consola. En el primer caso, también se da la posibilidad de trabajar con Neo4J de forma gráfica en lugar de hacerlo sólo mediante código. Para ello, proporciona una clase Java que implementa un servidor al que se accede desde un explorador web y que permite diversas operaciones como la visualización del grafo, un listado con las relaciones de cada nodo, realizar consultas utilizando Cypher o Gremlin o la creación y eliminación de índices.

Antes de descargar Neo4J, será necesario conocer qué necesitará el proyecto a implementar. Por ejemplo, si va a hacerse uso de tareas de monitorización o de copias de seguridad, la versión más sencilla (*Community*) no sería suficiente y habría que utilizar las versiones *Advanced* o *Enterprise*. Es importante tener esto presente ya que, mientras que la primera está basada en una licencia GPL<sup>21</sup>, las segundas hacen uso de AGPL<sup>22</sup>. Además de tener en cuenta la licencia, otro factor a considerar es el hardware mínimo requerido para que el trabajo con la base de datos tenga un rendimiento aceptable. Para facilitar esta

<sup>20</sup> Véase <http://www.neo4j.org/>

<sup>21</sup> Licencia de software más utilizada que permite a los usuarios la libertad de usar, compartir y modificar el software.

<sup>22</sup> Similar a las licencias GPL pero añadiendo la obligación de distribuir el software si éste se ejecuta para ofrecer servicios a través de una red de ordenadores.

tarea, Neo4J proporciona una herramienta en la que, a través del número de nodos, aristas y propiedades de cada uno de ellos calcula los requisitos mínimos de hardware para que la solución implementada sea lo más eficiente posible.

### 2.3.6 Selección del gestor

Con objeto de seleccionar el gestor a utilizar en el proyecto se instalaron y realizaron diversas pruebas con los gestores previamente mencionados.

En primer lugar, se descartó HyperGraphDB por el modelo utilizado. El hecho de emplear hiper-grafos complicaba el problema en lugar de facilitarlo puesto que no eran necesarias estructuras tan complejas.

Con los gestores restantes se tuvo en cuenta, en primer lugar, la cantidad de datos con la que se iba a trabajar y las licencias bajo las que trabajaba cada uno de ellos. Teniendo en cuenta esto, se eliminó el gestor DEX de la lista de posibilidades debido a que su versión gratuita sólo permitía el trabajo con un millón de objetos, cantidad que iba a ser sobrepasada por los conjuntos de datos con los que se iba a trabajar.

El siguiente factor a estudiar fueron los lenguajes de consulta con los que trabajaban los tres gestores restantes. Aunque todos ofrecían la posibilidad de utilizar Gremlin, se optó por asegurar la posibilidad de poder trabajar con otros lenguajes en caso de que la implementación de algunas consultas fuera demasiado compleja para su implementación en Gremlin. Puesto que tanto Neo4J como OrientDB ofrecían, además de Gremlin, lenguajes de consulta propios con una sintaxis más natural, se desechó el gestor InfiniteGraph, que tan sólo ofrecía el lenguaje de lógica de predicados.

Entre las dos últimas opciones se eligió, por su lenguaje SQL adaptado y el rendimiento que prometían sus características, el gestor OrientDB. Las desventajas de esta elección aparecieron cuando, al comenzar a trabajar con él, surgieron diferentes dudas y no se encontró suficiente información al respecto. Posiblemente debido a su juventud y a su pertenencia a una empresa pequeña, OrientDB aún no es lo suficientemente conocido como para encontrar soluciones a los problemas que pueden darse. Con este pensamiento práctico, y viendo las características y la gran cantidad de información y adeptos de Neo4J, se optó finalmente por este gestor.

### 3 Business Intelligence

En este capítulo se realiza una breve introducción al concepto *Business Intelligence*, se enumeran los beneficios que aporta, se describen las fases necesarias para el desarrollo de una solución de este tipo y se especifican las herramientas utilizadas en la realización de este proyecto.

#### 3.1 Introducción

En la actualidad, las empresas cada vez disponen de más información procedente de fuentes internas (sistemas corporativos, aplicaciones departamentales, etc.) y externas (INE, INEM, encuestas, redes sociales, etc.), lo que sumado a la competitividad y a la globalización complica sus posibilidades de adaptarse y sobrevivir a los cambios del entorno. Es por ello que las empresas deben encontrar la forma de anticiparse a la competencia y fidelizar clientes así como reducir y optimizar procesos. Estos objetivos hacen evidente la necesidad de acceder a la información de forma ágil y rápida.

Con el fin de ayudar al cumplimiento de estos objetivos, nació el *Business Intelligence* (BI). Su principal funcionalidad es la de apoyar a las empresas a la hora de mejorar su competitividad, lo que se consigue facilitando la información necesaria para la toma de decisiones. Este término fue utilizado por primera vez por Howard Dresner (1989) que lo definió como *“un conjunto de conceptos y métodos para mejorar la toma de decisiones utilizando para ello información sobre hechos pasados”*.

Otra definición de Business Intelligence (Gartner Group, 2006), lo define como *“un proceso interactivo para explorar y analizar información estructurada sobre un área, normalmente almacenada en un data warehouse, con el objetivo de descubrir tendencias o patrones, a partir de los cuales derivar ideas y extraer conclusiones. El proceso de BI incluye comunicar los descubrimientos y efectuar los cambios”*.

Por su parte, The Data Warehouse Institute define el BI como *“un término paraguas que abarca los procesos, las herramientas y las tecnologías para convertir datos en información, información en conocimiento y planes para conducir de forma eficaz las actividades de los negocios”*.

Todas las definiciones citadas recogen la misma idea: el objetivo fundamental de una solución Business Intelligence es la transformación de datos en información que pueda ser directamente utilizada para la toma de decisiones.

Las soluciones Business Intelligence aportan una serie de beneficios que pueden clasificarse, según Bill Whittlemore (2003) y Gabriel Fuchs (2003), en tangibles, intangibles y estratégicos:

- **Beneficios tangibles.** Entre estos se encuentran la reducción de costes, la generación de ingresos o la necesidad de emplear menos tiempo en algunas actividades. Esto se consigue, por ejemplo, identificando a los mejores proveedores para conseguir un precio menor, segmentando a los clientes o reasignando al personal de forma que se incremente la productividad.
- **Beneficios intangibles.** A partir de la información enfocada a la toma de decisiones, se mejorará este fin consiguiendo así mejorar la posición competitiva.

Esto se alcanza, entre otros, mejorando la atención al cliente aumentando así su satisfacción o facilitando el acceso a los datos mediante el uso de consultas, análisis o informes.

- **Beneficios estratégicos.** Identificar a qué clientes, mercados o productos se puede dirigir una campaña de marketing utilizando para ello tácticas como el análisis de estrategias de precios o la mejora de la toma de decisiones haciendo que sea más rápida, con más información y basada en hechos.

Pero no todo son ventajas: el desarrollo e implantación de soluciones Business Intelligence es costoso económicamente, debido principalmente al elevado coste de las licencias, y temporalmente (horas-persona) ya que estas soluciones involucran a bastantes personas de la organización en la fase de definición de métricas y durante la programación y validación de los procesos ETL, los cuales se describen en el apartado 3.2.3.

### 3.2 Fases de desarrollo de una solución BI

El desarrollo de una solución Business Intelligence sigue una serie de fases que se muestran en la Ilustración 3.1.

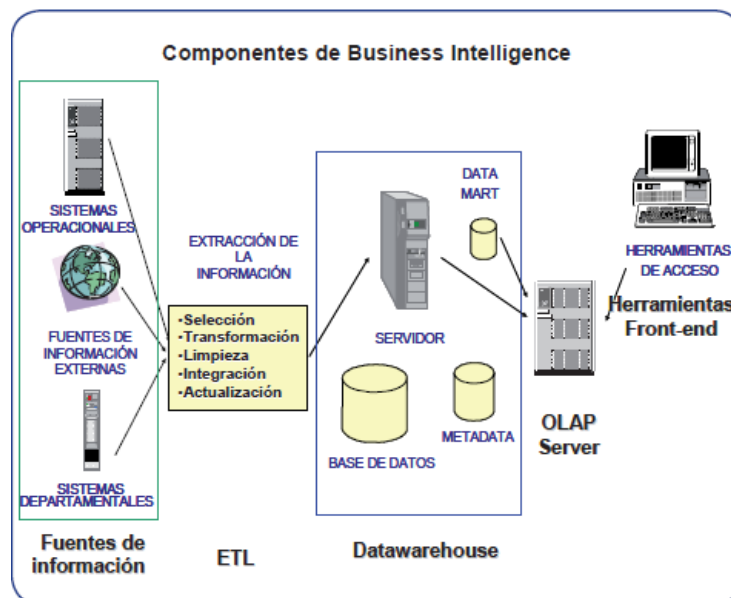


Ilustración 3.1 Componentes de una solución BI

En primer lugar se debe estudiar la problemática a la que la solución ha de dar respuesta. Conocido este objetivo, se analiza la información de la que dispone la empresa para responder a las preguntas del negocio. Esta información se debe trasladar y transformar antes de cargarse en el almacén de datos o *data warehouse* y, finalmente, la solución tiene que ofrecer al usuario final una interfaz para analizar y visualizar la información. Cada una de estas fases se detalla en los apartados siguientes.

#### 3.2.1 Definición de requisitos

En esta primera fase se definen las preguntas a las que la solución BI debe dar respuesta. Estas preguntas se conocen como métricas o indicadores claves de negocio y son la herramienta básica para saber si se están alcanzando los objetivos buscados. Para que cumplan las expectativas, deberán ser cuantificables de forma que esos objetivos se puedan medir.

A la hora de su definición, es fundamental tener en cuenta a los responsables de las diferentes áreas de la empresa con el fin de que faciliten la definición de dichos indicadores, su cálculo y qué valores reflejarían el éxito.

### 3.2.2 Fuentes de información

Una vez definidos los indicadores, se debe analizar la información de la que dispone la empresa. Esta información puede provenir de diversas fuentes:

- **Sistemas operacionales o transaccionales:** información interna de la empresa procedente de, entre otros, herramientas ERP<sup>23</sup> o CRM<sup>24</sup>.
- **Sistemas de información departamentales:** como pueden ser las previsiones, los presupuestos y las hojas de cálculo.
- **Fuentes de información externa:** estudios de mercado o información referente a la población, páginas web, redes sociales, ficheros procedentes de encuestas, etc.

Habitualmente, de estas fuentes se obtienen grandes cantidades de información por lo que es muy importante identificar si dicha información es apropiada en lo que refiere a formato, disponibilidad y calidad y si cumple las expectativas necesarias para responder a los indicadores definidos en el paso anterior. En caso contrario, se deberán modificar dichos indicadores para que se adapten a la información de la que se dispone o bien anular el cálculo del indicador. Una vez realizado este paso, es fundamental estudiar la calidad de los datos, aspecto del que se hablará en el apartado siguiente.

### 3.2.3 Proceso ETL

El proceso ETL (*Extraction, Transformation and Load*) tiene como objetivo extraer la información de las fuentes, transformarla y cargarla en el almacén de datos. En toda solución BI este proceso es clave y habitualmente consume entre un 60% y un 80% del tiempo total de desarrollo de la solución.

Este proceso, a su vez, puede dividirse en 5 subprocesos que se detallan a continuación:

1. **Extracción.** Consiste en extraer los datos de las diferentes fuentes de información. Tras este proceso, se puede decir que los datos están “en bruto”. Puede llevarse a cabo de forma “manual”, mediante lenguajes de programación como Java, o utilizando herramientas especializadas en ETL, con las que se podrá visualizar el proceso y detectar errores más rápidamente. Con frecuencia se hace necesario un almacén de datos intermedio, también llamado *data staging*, que suele utilizarse entre esta fase y las posteriores y que permanece oculto a los usuarios finales.
2. **Limpieza.** Es frecuente que los datos procedentes de las diversas fuentes de información no hayan sido depurados, es decir, que estén “sucios”. A menudo esta “suciedad” es producida por datos incompletos (por ejemplo, atributos en blanco o ausentes), con ruido (contienen errores como salarios negativos), e inconsistentes (edades que no coinciden con las fechas de nacimiento). Esta falta de calidad en los

---

<sup>23</sup> ERP (*Enterprise Resource Planning*) es un sistema de planificación de recursos empresariales que integra y maneja asuntos relacionados con operaciones de producción y distribución.

<sup>24</sup> CRM (*Customer Relationship Management*) es un sistema informático de apoyo a la gestión de las relaciones con los clientes en aspectos como las ventas y el marketing.

datos suele deberse, entre otros, a valores por defecto, errores humanos y modelos de datos no normalizados. Si se desea una solución en la que los resultados mostrados sean coherentes y fiables, es necesario que los datos estén “limpios”, esto es, que sean datos de calidad. Para ello, deberán ser precisos, íntegros, coherentes, completos, válidos, disponibles y accesibles. Este punto de la transformación es de vital importancia ya que, como afirma Bill Inmon (2006), *“las organizaciones actúan bajo la suposición de que la información de la que disponen es precisa y válida. Si la información no es válida, entonces no pueden responder a las decisiones basadas en ella”*.

3. **Transformación.** Una vez que los datos están “limpios”, el siguiente paso es transformarlos mediante cambios de formato, sustitución de códigos y cálculo de valores derivados y agregados (estos últimos habitualmente se calculan y se almacenan para mejorar el rendimiento de las consultas realizadas sobre el almacén de datos). En esta etapa también se incluye la definición del nivel de grano o detalle, concepto que se definirá en el apartado siguiente.
4. **Integración.** En este momento los datos se cargan en el almacén de datos. Es fundamental que este paso se lleve a cabo correctamente.
5. **Actualización.** Será necesario definir con qué periodicidad se realizarán nuevas cargas de datos al almacén.

Para llevar a cabo todas estas tareas que se acaban de definir, existen herramientas ETL que ayudan en su ejecución y de las cuales se hablará en el apartado 3.3.

#### 3.2.4 Almacén de datos o *data warehouse*

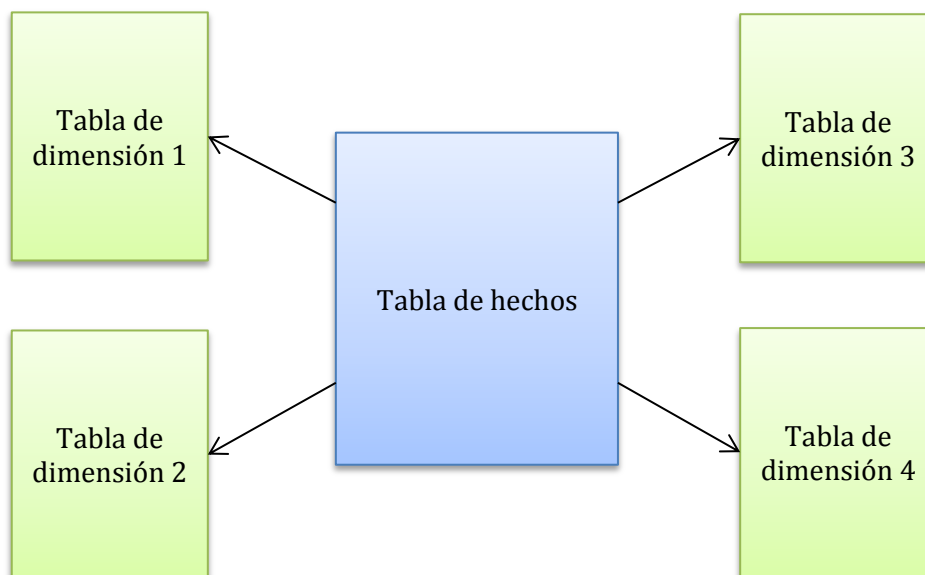
Los almacenes de datos o *data warehouse* (DW) surgieron como respuesta a las necesidades de los usuarios, que necesitan información consistente, integrada, histórica y preparada de forma que pueda ser analizada para la toma de decisiones.

Desde su aparición, han sido definidos de diferentes formas. Una es la realizada por Hugh J. Watson (2006), quien lo definió como *“una colección de información creada para soportar las aplicaciones de toma de decisiones”*. Sobre esta definición, Bill Inmon (2005) añadió las características que debía cumplir, siendo éstas *“estar orientado sobre un área, integrado, indexado al tiempo y no volátil de forma que soporte la toma de decisiones”*. Analizando esta definición, se deduce que un almacén de datos deberá estar enfocado a resolver un problema de negocio, para lo cual la información deberá transformarse de forma que esté estandarizada y pueda ser útil. Además, esta información deberá almacenarse refiriéndose a medidas de tiempo (en horas, días, semanas, meses, etc.) y su actualización se hará periódicamente de forma preestablecida.

Otro autor, Ralph Kimball (2002) lo definió como *“el lugar donde se publica la información”* e indicó los objetivos que debe cumplir. Entre estos objetivos están su alcanzabilidad (pudiendo ser corporativa o departamental), la consistencia de la información que almacena, la posibilidad de analizarla por separado o combinándola y la disponibilidad de herramientas de consulta, análisis y presentación.

Puesto que un almacén de datos es una base de datos, será necesario diseñar las distintas tablas y relaciones que la forman. Con este fin, se estudiarán las cuestiones planteadas inicialmente y a las que la solución BI debe dar respuesta. Cada almacén sigue

un esquema multidimensional, generalmente basado en “estrellas”, adaptado a cada problemática y que da respuesta a las preguntas planteadas. En este esquema se distinguen dos tipos de tablas: las tablas de hechos y las tablas de dimensiones. Las primeras almacenan aquello que se desee medir o analizar (es decir, los “hechos”), mientras que las segundas indican cómo se quieren medir o, lo que es lo mismo, permiten agrupar los hechos de la tabla de hechos en función de los valores de la dimensión. El esquema de la base de datos resultante se muestra en la Ilustración 3.2.



**Ilustración 3.2** Esquema multidimensional en estrella

A grandes rasgos, los pasos que se deben seguir en el diseño de un almacén de datos son los que se describen a continuación:

1. **Elegir la granularidad.** En función de las preguntas que se deseen responder, será necesario un grano (o nivel de detalle) más o menos fino. El nivel de granularidad condiciona la flexibilidad de las consultas por lo que es muy importante tener en cuenta que, una vez que los datos estén guardados, no se podrá entrar en más detalle. Por ello, lo más recomendable es almacenarlos al máximo nivel de detalle posible aunque en la práctica no se utilice.
2. **Decidir las dimensiones.** La granularidad elegida en el punto anterior determina las dimensiones, por lo que habrá que ajustarlas al detalle elegido. Además, los atributos de cada dimensión servirán para definir las agregaciones. Aunque para cada solución BI las dimensiones serán diferentes, todas tendrán una dimensión en común: la dimensión temporal. Esta dimensión, como su propio nombre indica, almacenará fechas y, si el problema requiere que se almacene el instante del día, también será necesaria una dimensión para las horas. Lo mejor es que el grano de la dimensión temporal sea diario: de esta forma, podremos agrupar los datos por días, semanas, meses, etc. y, además, disponer de información más precisa como, por ejemplo, qué día de la semana es, si es festivo o si corresponde a un periodo vacacional.

3. **Determinar los hechos.** En este punto se decidirá qué hechos se van a implementar, es decir, qué medidas formarán la tabla de hechos. Para trabajar mejor, es conveniente que estas medidas puedan sumarse entre sí y, en caso de no ser posible, se podría trabajar con ellas utilizando valores medios.

Hasta ahora se han descrito los pasos a seguir para diseñar un almacén de datos sencillo pero habitualmente estas bases de datos requieren la integración de varias estrellas que trabajan con dimensiones a distinto nivel de detalle, por lo que es necesaria su normalización (o jerarquización) obteniendo el llamado esquema en copo de nieve (*snowflake*). Esta situación se da cuando, por ejemplo, se almacena para cada usuario datos referentes su nivel de estudios o la ciudad en la que vive. El nivel de estudios y la ciudad se encontrarían almacenados en dimensiones jerarquizadas a las que se haría referencia en la dimensión del usuario. Otra característica que es habitual encontrar es la presencia de dos tipos de tablas de hechos: transaccionales y *snapshots*. Las primeras están formadas por filas que representan eventos mientras que las segundas corresponden a capturas de la base de datos en un instante de tiempo determinado. Mientras que las transaccionales tienen un mayor nivel de detalle, los *snapshots* presentan una visión acumulativa sin entrar en detalle pero son útiles para conocer la evolución de la información.

Uno de los aspectos fundamentales de un almacén de datos es su actualización periódica, para lo cual habrá que definir procesos ETL y un *data staging* que permita conocer los cambios en las fuentes de información originales para trasladarlos al almacén. Asimismo, al tratarse de una base de datos, habrá que vigilar que la información esté siempre disponible, que se cumplan ciertos requisitos de rendimiento y realizar (en plazos fijados previamente) copias de seguridad y recuperación.

### 3.2.5 Explotación de los datos

Por último, una solución BI deberá proporcionar herramientas para analizar y visualizar la información que reside en el almacén. Existen diferentes herramientas, cada una con un propósito distinto y que se detallan a continuación.

- **Generadores de informes.** Utilizadas por desarrolladores, su función es la creación de informes estándar para ciertos grupos de usuarios, departamentos o la organización.
- **Herramientas de usuario final de consultas e informes.** Son empleadas por los usuarios finales y no necesitan programación. Dan al usuario la posibilidad de crear informes.
- **Herramientas OLAP (On-Line Analytical Processing).** Con ellas, los desarrolladores pueden construir cubos OLAP y los usuarios finales pueden tratar la información de forma multidimensional puesto que permiten explotar la información teniendo en cuenta diferentes perspectivas y periodos de tiempo.
- **Herramientas de *dashboard* y *scoreboard*.** Estas herramientas permiten al usuario ver información de manera más sencilla mediante el uso de recursos gráficos a la vez que ofrecen la posibilidad de entrar en más detalle y en informes.
- **Herramientas de planificación, modelización y consolidación.** Dan la posibilidad al usuario final de crear planes de negocio y simulaciones a partir de la información existente y suelen utilizarse, entre otros, para la elaboración de presupuestos y previsiones.



- **Herramientas *data mining*.** Permiten la creación de modelos estadísticos de las actividades. El *data mining*, o minería de datos, tiene como objetivo el descubrimiento de patrones presentes en la información y sus resultados suelen emplearse en tareas como la segmentación, la clasificación o las previsiones.

Entre estas herramientas mencionadas cabe destacar las herramientas OLAP debido a su gran importancia en el mundo del Business Intelligence. Con el objetivo de que los usuarios vean la información agregada a diferentes niveles y sobre diferentes dimensiones (por ejemplo, el número de compras realizadas por clientes con edades comprendidas en un cierto rango y de un país determinado) nació OLAP. La forma de representar las herramientas OLAP son cubos cuyas aristas se corresponden con las diferentes dimensiones que participan en la consulta tal como se muestra en la Ilustración 3.3, donde puede verse el número de libros que compró un cliente determinado en un año dado. Esta información se recoge en los cubos individuales que forman el cubo y que son lo que se había mencionado anteriormente como “hechos” en la tabla de hechos.

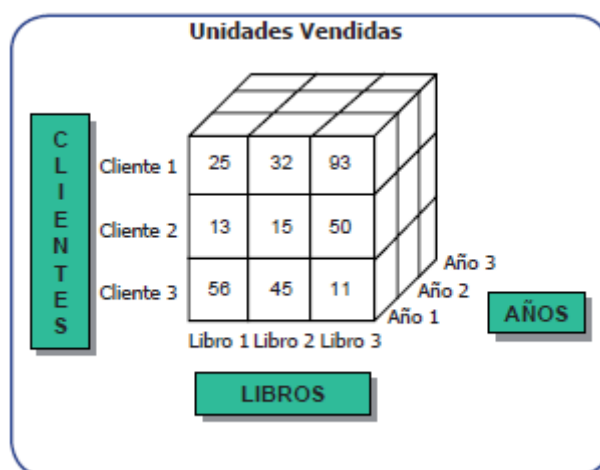


Ilustración 3.3 Cubo OLAP

Con el fin de agilizar las consultas, las herramientas OLAP permiten la realización de ciertas operaciones exclusivas de los cubos y pueden almacenarse siguiendo diferentes opciones de almacenamiento, como MOLAP si lo que se busca es rendimiento, ROLAP en caso de que la capacidad de almacenamiento que se necesite sea grande, HOLAP si se desea combinar ambas y DOLAP si se necesita trabajar con MOLAP en un equipo cliente.

### 3.3 Herramientas comerciales BI

Para la realización de todas las fases que conlleva una solución BI es recomendable el uso de herramientas orientadas al desarrollo de estas soluciones. Estas herramientas deberán ajustarse a la demanda del mercado, que reclama la aceleración de los procesos de extracción y carga de datos, la posibilidad de trabajar con diversos formatos de fuentes y datos, la capacidad de soportar grandes complejidades y la cercanía a las cargas en tiempo real. Aunque hay una gran variedad en el mercado, tanto en software *open-source* (Pentaho<sup>25</sup> o Palo<sup>26</sup>) como comercial (Oracle Business Intelligence<sup>27</sup>) en el presente

<sup>25</sup> Véase <http://www.pentaho.com>

<sup>26</sup> Véase <http://www.palo.net>

<sup>27</sup> Véase <http://www.oracle.com/es/solutions/midsized/oracle-products/business-intelligence/index.html>

apartado se describen las herramientas empleadas para la realización de este proyecto y que pertenecen al entorno Microsoft SQL Server Business Intelligence Development Studio.

Business Intelligence Development Studio (BIDS) es el entorno de desarrollo implementado por Microsoft e integrado en Visual Studio. Se utiliza para el desarrollo de herramientas orientadas al análisis de datos y soluciones Business Intelligence, para lo que hace uso de tres plataformas: Integration Services, Reporting Services y Analysis Services. Aunque está basado en el entorno de desarrollo Visual Studio, proporciona extensiones y tipos de proyecto específicos para Business Intelligence tales como herramientas para la creación de informes, flujos de datos para el proceso ETL, cubos OLAP y estructuras para realizar minería de datos.

El primero de los componentes de este entorno mencionado en el párrafo anterior es SQL Server Integration Services (SSIS), una plataforma dirigida a la integración de datos y al flujo de trabajo, es decir, se encarga de facilitar el proceso ETL y, además, permite automatizar las actualizaciones de las bases de datos y de los cubos. Realizar una carga de datos es tan sencillo como definir el origen y el destino, pudiendo ser éstos ficheros de texto o instancias SQL y dando la posibilidad al usuario de hacerlo de manera gráfica mediante *drag-and-drop*<sup>28</sup>. Además, esta carga de datos se puede mejorar puesto que la plataforma ofrece las características que se describen a continuación.

- **Conexiones.** Cada conexión deberá incluir la información necesaria para conectarse a una fuente de datos concreta, pudiendo ser esta fuente, entre otros, un fichero, un cubo OLAP o una base de datos relacional.
- **Tareas.** Actúan de manera atómica e implementan alguna acción, pudiendo elegir entre 24 tareas que van desde la copia de datos hasta la transformación de estos, lo cual es de vital importancia para todo proceso ETL.
- **Restricciones de precedencia.** Las tareas se unen entre sí mediante estas restricciones y deciden si una tarea puede ejecutarse o no. Si, por ejemplo, varias tareas se ejecutan en paralelo, sólo podrán realizarse si estas restricciones lo permiten. Además, son de utilidad para definir caminos de ejecución en caso de que, por ejemplo, se produjera un error.
- **Manejadores de eventos.** Permiten ejecutar tareas en respuesta a ciertos eventos que pueden ocurrir.
- **Variables.** Cada tarea puede hacer referencia a variables para almacenar resultados, tomar decisiones o configurarse de una forma u otra.

Estas herramientas utilizadas para la carga de datos se integran en lo que la plataforma llama paquete, cuyo contenido se almacena en lenguaje XML aunque se visualiza gráficamente. Habitualmente, para cargar un almacén de datos, serán necesarios varios paquetes pudiendo definirse en qué orden se han de ejecutar mediante las restricciones de precedencia mencionadas anteriormente. Además, a la hora de cargar los datos en el almacén, es muy probable que estos requieran una transformación, para lo que se proporcionan tareas para el flujo de datos como agrupaciones, búsquedas, columnas derivadas, ordenación, uniones, etc.

---

<sup>28</sup> Expresión que se refiere a la acción de mover objetos gráficos utilizando el ratón.

Aunque las características que se ofrecen de modo gráfico y que se han definido anteriormente son suficientes para realizar una carga compleja, también se ofrece la posibilidad de definir nuevos objetos mediante código de programación.

La siguiente plataforma mencionada fue SQL Server Reporting Services (SSRS). Esta plataforma tiene como finalidad la creación y distribución de informes interactivos en los que visualizar la información, pudiendo ser generados en varios formatos como Excel, PDF, XML o HTML. Proporciona además una aplicación web para trabajar con los informes, para lo que utiliza un servidor de informes integrado en la plataforma. Con esta aplicación web los usuarios pueden ver y modificar informes a la vez que configuran aspectos de seguridad. Aquí, la seguridad está basada en roles, pudiendo asignarse diferentes niveles de seguridad a cada informe, a cada grupo de informes o al proyecto completo.

Finalmente, la última plataforma incluida en este entorno es SQL Server Analysis Services (SSAS). Su función es la del procesamiento analítico en línea (OLAP), mencionado en el apartado 3.2.5, y la minería de datos. Se emplea para analizar información que puede proceder de varias bases de datos y tablas. Para almacenar dicha información, proporciona la opción de utilizar MOLAP, ROLAP o HOLAP en función de las necesidades. A la hora de definir y trabajar con los cubos, Analysis Services ofrece diferentes modelos de objetos (XML, OLE DB, ADO .NET, etc.) y de lenguajes de consulta (XML, MDX, LINQ, SQL y DMX).

La elección de Microsoft SQL Server Business Development Studio se hizo, además de por las facilidades que ofrece, por su integración en SQL Server: tan sólo es necesaria la instalación de Microsoft SQL Server para disponer de ella, lo que la convierte en la herramienta comercial para BI más económica.

## 4 Especificación funcional

El contenido de este capítulo se divide en dos partes: la primera, tiene como objetivo la definición de los indicadores que se proponen clasificados en tres categorías: alcanzabilidad, compromiso e influencia; y la segunda, se centra en describir en detalle los tres conjuntos de datos empleados en este proyecto y los indicadores que, por sus características y naturaleza, se les aplican.

### 4.1 Requisitos funcionales: métricas

Los requisitos funcionales se utilizan para definir las funciones que debe llevar a cabo la aplicación software. En este proyecto, los requisitos funcionales se resumen exclusivamente en las métricas o indicadores a los que la solución final deberá dar respuesta a través de informes.

Con el fin de mejorar la comprensión de este apartado se muestra, en la Ilustración 4.1, la imagen de un grafo genérico que resume los nodos y las relaciones más importantes a partir de los cuales se han definido las métricas.

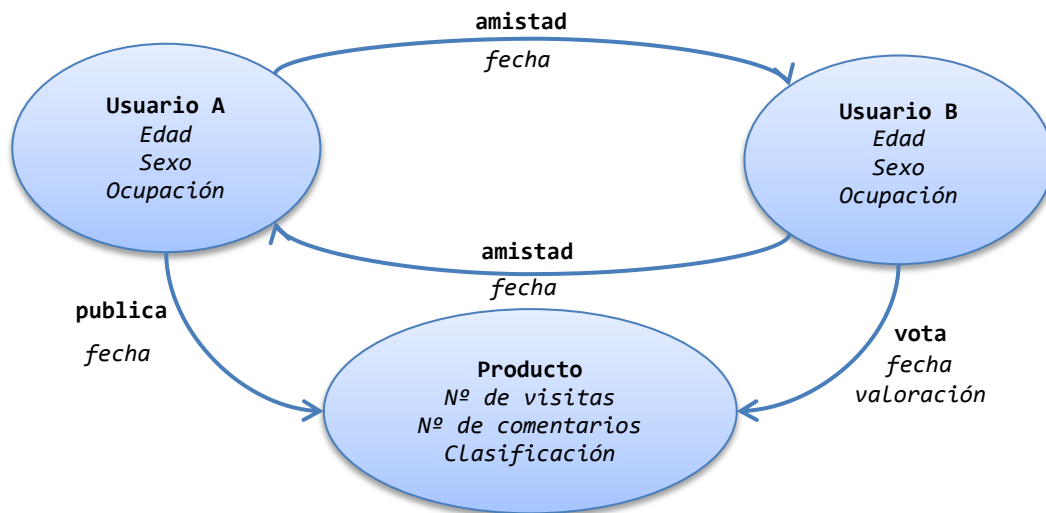


Ilustración 4.1 Grafo genérico

Como se observa en la imagen, aparecen como nodos los usuarios y los productos (pudiendo ser estos una fotografía, un artículo, etc. y de los que se conocen ciertos datos) y, como relaciones, la amistad entre usuarios (de la que se conoce la fecha en que se produjo), que pueden ser recíprocas o no, y las relaciones entre usuarios y productos, pudiendo ser éstas de dos tipos: la publicación de un producto por parte de un usuario (incluyendo la fecha en la que se publicó) y el voto de un usuario a un producto (del que se dispone de la fecha y la valoración dada).

Teniendo este esquema de red social general, en los siguientes apartados se describen las métricas que definen los requisitos funcionales del proyecto, agrupadas en métricas de alcanzabilidad, de compromiso y de influencia (Palazuelos & Zorrilla, 2012).

#### 4.1.1 Métricas de alcanzabilidad

Las métricas de alcanzabilidad son aquellas que, en conjunto, permiten conocer el grado de propagación efectiva de cierto contenido. Si, por ejemplo, una empresa lanza un nuevo producto o servicio al mercado le interesará conocer qué valoración está recibiendo su nuevo producto en el medio (en este caso, en una red social) de forma que pueda utilizar esa información para mejorar o cambiar la forma de publicitar dicho producto o el producto en sí mismo.

Esta información puede obtenerse a partir del análisis de los siguientes indicadores:

- **Crecimiento de usuarios.** Incremento de los usuarios conectados a través de la red social como consecuencia del lanzamiento de un producto al mercado. Esto es:

$$\frac{N^{\circ} \text{ de usuarios a fecha } Y - N^{\circ} \text{ de usuarios a fecha } X}{N^{\circ} \text{ de usuarios a fecha } Y}$$

**Ecuación 4.1** Crecimiento de usuarios

Donde el usuario de la aplicación implementada será el encargado de elegir la fecha de inicio ( $X$ ) y la fecha de fin ( $Y$ ).

- **Crecimiento de usuarios activos.** Este indicador se calcula del mismo modo que el anterior pero se tienen en cuenta sólo los usuarios “activos”, esto es, aquellos que han realizado actividades en la red social, como votar o publicar productos, durante el período de análisis.
- **Velocidad de propagación.** Se calcula como el tiempo que transcurre (minutos, horas o días) desde que un producto es publicado hasta que es votado por primera vez.
- **Actividad en la red.** Contabiliza el número de productos publicados, votos realizados o nuevas amistades que se han producido en la red en un determinado período de tiempo elegido por el usuario de la interfaz.
- **Frecuencia de publicación de los usuarios en la red.** Tiene como objetivo conocer cada cuánto tiempo se publican nuevos productos. Es decir, calcular qué número de fotografías o artículos son publicados por los usuarios en un periodo de tiempo seleccionado desde la interfaz del cliente. Por ejemplo, si la fecha elegida comprende el último mes uno de los posibles resultados es que se hayan publicado dos fotografías en ese tiempo. Para su cálculo se hace uso de la siguiente ecuación:

$$\frac{N^{\circ} \text{ de productos publicados en ese periodo}}{\text{Último día, semana o mes del periodo} - \text{Primer día, semana o mes del periodo}}$$

**Ecuación 4.2** Cálculo de la frecuencia

- **Frecuencia de voto.** Se calcula con la Ecuación 4.2 pero sustituyendo el número de productos por el número de votos que el usuario realiza sobre los productos en el periodo seleccionado.
- **Proximidad.** Indicador que muestra qué porcentaje de las amistades que se dan en la red social se encuentra a distancia 2, 3, 4, etc. en un periodo de tiempo elegido por el usuario de la interfaz.

#### 4.1.2 Métricas de compromiso

La finalidad de estas métricas es la de conocer el grado de participación e implicación de los usuarios.

- **Crecimiento de seguidores del perfil del usuario.** Consiste en conocer cuántos nuevos amigos o seguidores consigue el usuario en un periodo de tiempo dado. Para ello, se seguirá nuevamente la Ecuación 4.1, pero sustituyendo el número de usuarios por el de seguidores.
- **Tiempo que el usuario pasa conectado a la red.** Realiza un promedio del tiempo que un usuario está conectado cada uno de los días pertenecientes a un período de tiempo dado.
- **Número de visitas que recibe su perfil.** Se encarga de contabilizar el número de visitas que ha recibido el perfil de un usuario en el periodo de análisis dado.
- **Número de votos y/o comentarios recibidos.** Contabiliza el número de votos y/o comentarios que han recibido los productos de un usuario en el periodo de análisis.
- **Reciprocidad.** De utilidad en redes dirigidas, en las que ser amigo o seguidor de un determinado usuario no implica que esa relación exista en sentido contrario. Para el cálculo de este indicador una estructura en grafo resulta de gran utilidad: basta con obtener una arista saliente que represente una amistad entre el nodo origen y el nodo destino y comprobar si también existe una arista del nodo destino al nodo origen. De esta forma, se puede comprobar (tanto por usuario como en la totalidad de la red) qué porcentaje de las relaciones de amistad son recíprocas.
- **Fortaleza de los enlaces.** Para usuarios entre los que existan relaciones (por ejemplo, autores que publican artículos juntos) se calculará la fortaleza del enlace como un ratio entre el número de artículos publicados juntos con respecto al total de las publicaciones realizadas por ambos en un periodo de análisis dado. Por ejemplo, si el autor *A* y el autor *B* han publicado 2 artículos juntos y el autor *A* ha publicado en total 3 y el autor *B* 7, el ratio sería 2/10 (esto es, los dos artículos que han publicado juntos dividido entre la suma de todos sus artículos publicados). Esto queda recogido en la Ecuación 4.3.

$$\frac{N^{\circ} \text{ de productos publicados juntos}}{N^{\circ} \text{ de productos publicados por } A + N^{\circ} \text{ de productos publicados por } B}$$

**Ecuación 4.3** Cálculo de la fortaleza de los enlaces

- **Fidelidad de amigos.** Mediante el uso de este indicador se podrá conocer qué usuarios de la red social tienen amigos más fieles. Para obtener cómo de fiel es un amigo (*A*) de un determinado usuario (*B*) se hallará el número de votos que el usuario *A* realiza sobre las fotos del usuario *B* respecto al total de sus votos, es decir, si *A* realiza en total 5 votaciones siendo 2 de ellas sobre los productos publicados por *B*, su ratio de fidelidad será de 2/5. Este cálculo sigue la Ecuación 4.4.

$$\frac{N^{\circ} \text{ de votos del usuario } A \text{ sobre productos de } B}{N^{\circ} \text{ total de votos emitidos por } A}$$

**Ecuación 4.4** Cálculo de la fidelidad de un usuario amigo

### 4.1.3 Métricas de influencia

Con estas métricas se trata de calcular qué influencia o grado de movilización puede causar un usuario sobre otros.

- **Número de votos recibidos de sus amigos y de los amigos de sus amigos.** Para cada producto publicado por un usuario, se calculará el número de votos recibido de sus amigos directos (usuarios con proximidad 1), los amigos de sus amigos (usuarios con proximidad 2) y así sucesivamente, de forma que se conozca no sólo la distancia sino también el peso de esa distancia, siendo este peso el ratio entre el número de votos recibidos con las diferentes proximidades entre el número total de votos recibidos.
- **Número de usuarios que votan o comentan un producto después de que uno de sus amigos les envíe un mensaje.** Se contabilizará cuántos de los mensajes enviados por un usuario a sus amigos con el objetivo de que voten o comenten un producto consiguen que se realicen dichas acciones sobre el producto. Se deberá decidir cuánto tiempo puede transcurrir desde que se envía el mensaje hasta que el receptor vota o comenta un producto de forma que si, por ejemplo, transcurre un mes, no se tenga en cuenta.
- **Número de usuarios que se dan de alta en la red social tras recibir la invitación de un amigo.** Para cada usuario se calculará cuántas de las invitaciones enviadas tienen como resultado el alta del usuario al que envió la invitación, pudiendo elegir diferentes periodos de tiempo para el análisis.
- **Usuarios líderes.** Señala quiénes son usuarios líderes, esto es, aquellos que por su actividad agrupan a un conjunto amplio de usuarios. Es decir, son los usuarios que representan diferentes grupos en la red (Palazuelos & Zorrilla, 2011).

## 4.2 Conjuntos de datos

Para llevar a cabo este proyecto se han utilizado tres conjuntos de datos de diferente naturaleza y características que permiten responder a la mayoría de las métricas definidas en el apartado 4.1. Debido a sus diferencias, no todos los conjuntos de datos permiten responder a las mismas métricas. En las siguientes líneas se describirán los conjuntos utilizados así como las métricas que se les han podido aplicar.

### 4.2.1 MovieLens

Creado por el grupo de investigación GroupLens<sup>29</sup>, este conjunto de datos recoge información procedente de la página MovieLens<sup>30</sup>, en la que los usuarios votan a diferentes películas con una valoración comprendida entre 1 y 5, recogándose la fecha y hora en la que se realizó la votación. De los usuarios se conoce su sexo, edad, ocupación y código postal, y de las películas su título, su fecha de estreno y los géneros cinematográficos a los que pertenecen.

Estudiando las características de este conjunto, las métricas que se le pueden aplicar son las que se describen a continuación.

---

<sup>29</sup> Véase <http://www.grouplens.org>

<sup>30</sup> Véase <http://movielens.umn.edu>

- **Métricas de alcanzabilidad**

- **Frecuencia de voto:** puesto que se conoce la fecha y la hora en la que se produce cada votación, es posible calcular la frecuencia de voto de cada usuario por día, semana, o mes, esto es, calcular el número de votos que realiza el usuario por día, semana y mes. Además, y haciendo uso de los datos demográficos de los usuarios, existe la posibilidad de conocer esta frecuencia a través de diferentes perspectivas como la edad, el sexo, la ocupación o el área geográfica del usuario. Para su cálculo se empleará la Ecuación 4.2.

- **Métricas de compromiso**

- **Crecimiento de votos:** siguiendo la Ecuación 4.1 y sustituyendo el número de usuarios por el de votos, a partir de una fecha de inicio y una fecha de fin se podrá medir tanto el crecimiento de votos por usuarios (es decir, el crecimiento de votos emitido por cada usuario) como por películas (crecimiento de votos recibidos). Además, se podrá mostrar dicho crecimiento por días, semanas, meses, trimestres o años así como agrupándolo según los datos demográficos de los usuarios o por el género de las películas.
- **Número de votos y valoración media:** para cada película es posible obtener el número de votos recibidos y la valoración media de estos. Asimismo, y puesto que se conoce la fecha en la que se produjeron dichos votos, es posible comparar el número de votos y la valoración media en dos periodos diferentes de fechas así como agrupar estos por semana, mes o trimestre. De igual forma se podrán conocer los géneros favoritos de los usuarios agrupando el número de votos por el género de las películas así como las preferencias de los usuarios según sus datos demográficos (edad, sexo o nivel de estudios).

Además, aprovechando la infraestructura construida para calcular estas métricas, también se puede dar respuesta a preguntas tales como qué diez usuarios son los más activos, qué días y a qué horas se produce mayor actividad en la red social o qué películas reciben un mayor número de votos.

#### 4.2.2 Flickr

Este conjunto de datos contiene parte de la actividad registrada en la red social de fotografía Flickr en un periodo de tiempo comprendido entre el 2 de noviembre de 2006 y el 18 de mayo de 2007, proporcionándose además capturas o *snapshots* que recogen un resumen de la actividad realizada en tres periodos de tiempo diferentes. El primero de ellos recoge la actividad desarrollada del 2 de noviembre de 2006 al 3 de diciembre de 2006, el segundo del 4 de diciembre de 2006 al 2 de febrero de 2007 y el tercero del 3 de febrero de 2007 al 18 de mayo de 2007. Con el fin de agilizar la carga de datos y la ejecución de consultas, y aunque se ha hecho uso de los tres *snapshots* mencionados, tan sólo se han utilizado los datos correspondientes a la actividad detallada del primer periodo de tiempo (del 2 de noviembre al 3 de diciembre). El detalle de estos datos recoge información referente a las relaciones de amistad entre usuarios, las cuales no son recíprocas y de las que se conoce, además de los usuarios implicados, la fecha en la que se produjeron, datos relativos a la carga de fotografías (usuario propietario y la fecha y hora



de la carga), información sobre las votaciones de fotografías (qué usuario marca una fotografía como favorita y en qué día y hora lo hace), y datos sobre las fotografías (número de visitas y comentarios recibidos).

Por su parte, los *snapshots* mencionados recogen información referente a la actividad de cada usuario en un periodo de tiempo, siendo esta información el número de fotografías que ha publicado, el número de fotografías que ha marcado como favoritas, su número de amigos (o usuarios que sigue) y el número de amigos de sus amigos.

Con la información de la que se dispone, las métricas que es posible calcular son:

- **Métricas de alcanzabilidad**

- **Crecimiento de usuarios:** aunque no se dispone de información referente a la fecha en la que cada usuario se registra en la red social, esta información puede extraerse de los *snapshots* proporcionados puesto que el número de usuarios varía de uno a otro y corresponden a periodos de tiempo distintos. Para su cálculo se hace uso de la Ecuación 4.1.
- **Crecimiento de usuarios activos:** igual a la métrica anterior pero tan sólo se contabilizan aquellos usuarios que hayan, como mínimo, publicado una fotografía, votado una fotografía y establecido una relación con otro usuario.
- **Velocidad de propagación:** se calcula como la diferencia temporal (minutos, horas, días, etc.) desde que un usuario publica una fotografía hasta que ésta es votada por primera vez.
- **Actividad:** contabiliza, en un periodo de tiempo elegido, el número de votos que ha recibido cada fotografía o cada usuario, el número de votos emitido por un usuario o el número de nuevos amigos o seguidores de un usuario.
- **Frecuencias de voto, de publicación y de amistad:** se obtendrá, por cada usuario, cuántos votos realiza, cuántas fotografías publica y cuántos nuevos amigos hace por día, semana y mes en un periodo de tiempo dado haciendo uso de la Ecuación 4.2.
- **Proximidad:** por cada relación de amistad se conoce la distancia existente entre los usuarios implicados antes de que se estableciera dicha amistad. Esta medida también se aplica a las votaciones que los usuarios realizan sobre las fotografías, donde se calcula cómo de próximos se encuentran el usuario que vota y el propietario de la imagen. Hallando qué porcentaje de estas relaciones se encuentra a diferentes proximidades, se obtendrá una clasificación que muestre cada valor de la proximidad junto con su porcentaje para conocer, a nivel de red social, cómo de cohesionada se encuentra la red pudiendo comparar estos valores por periodos de tiempo.

- **Métricas de compromiso**

- **Crecimiento de fotografías, votos y seguidores:** para cada usuario se calculará la diferencia de estos valores (fotografías subidas, votos recibidos y nuevos seguidores) entre dos fechas sustituyendo a los usuarios por dichos valores en la Ecuación 4.1.

- **Número de votos, visitas y comentarios:** con los datos de las fotografías subidas por un usuario se hallará el número total de votos, visitas y comentarios que han recibido sus fotografías.
- **Reciprocidad:** puesto que las relaciones de amistad son dirigidas, es decir, que un usuario  $X$  sea amigo o seguidor del usuario  $Y$  no significa que exista la misma relación desde el usuario  $Y$  al usuario  $X$ , se obtendrá si la relación existe en ambos sentidos. Con esto, se podrá obtener qué tanto por ciento del total de las relaciones de amistad son recíprocas y cuáles no (tanto por usuario como en la totalidad de la red) y, además, se podrá indicar un periodo a estudiar.
- **Fidelidad de amigos:** después de que se establezca una relación de amistad entre usuarios, se contabiliza la fidelidad como el ratio entre el número de votos que un usuario realiza sobre las fotografías de su usuario amigo dividido entre el número total de votos que ha realizado, tal como indica la Ecuación 4.4.
- **Métricas de influencia:**
  - **Número de votos recibidos de sus amigos y de los amigos de sus amigos:** con el fin de conocer la influencia de cada usuario, se calcula qué porcentaje de votos sobre las fotografías que ha publicado han sido realizados por usuarios que se encuentran a proximidad 1, 2 o mayor que 2. Puede limitarse por tiempo, de forma que se pueda comparar si dicho valor varía entre dos fechas.

Al igual que ocurre con MovieLens, con Flickr también se puede obtener qué usuarios son más activos a la hora de votar o publicar fotos, a qué hora del día se realiza la mayor carga de fotografías o qué días se registra una mayor actividad en la red. Asimismo, haciendo uso del número de votos, visitas y comentarios, se puede calcular no sólo la suma de cada uno de estos valores recibidos por las fotografías de cada usuario, sino también el número máximo de votos, visitas o comentarios recibidos teniendo en cuenta todas las fotografías publicadas por un usuario.

#### 4.2.3 DBLP

DBLP recoge información referente a artículos relacionados con la computación y publicados en un periodo de tiempo comprendido entre el año 1936 y el 2011. De cada publicación se conoce su título, su clasificación, el año en el que se publicó, las publicaciones a las que cita y sus autores, de los cuales se tiene su nombre completo.

En el caso de este conjunto de datos, las relaciones de coautoría entre usuarios o autores se consideran relaciones de amistad, de forma que la primera vez que dos autores realizan juntos una publicación se establece una relación de amistad y, a partir de ese momento, dichos autores se consideran “amigos”. Otro aspecto a tener en cuenta son las citas, puesto que la mayor parte de las publicaciones recogidas tienen referencias a otras publicaciones. Estas referencias se interpretarán como relaciones de voto, siendo el usuario votante el autor que escribe la publicación que cita y el producto votado la publicación que es citada.

Utilizando esta información, las métricas a las que puede dar respuesta son las siguientes:

- **Métricas de alcanzabilidad**

- **Velocidad de propagación:** se calculará como el tiempo que transcurre (pudiendo ser éste tan sólo años debido a la información proporcionada) desde que un artículo es publicado hasta que es citado por primera vez.
- **Frecuencia de publicación:** de cada autor se calculará cuántas publicaciones realiza por año (sólo puede calcularse la frecuencia por año debido a las características del conjunto de datos) haciendo uso de la Ecuación 4.2. Puede obtenerse estableciendo un período de tiempo, de forma que sólo se tengan en cuenta las publicaciones realizadas entre esos años dados.
- **Frecuencia de citas:** de las publicaciones escritas por un autor se calculará, haciendo uso nuevamente de la Ecuación 4.2, cuántas citas realiza por año en estas publicaciones.
- **Proximidad:** se calcula la distancia a la que se encontraban los autores antes de escribir un artículo juntos por primera vez de forma que se pueda obtener qué tanto por ciento de los coautorías tienen diferentes proximidades. De igual forma, puede obtenerse este valor para las citas en los artículos. Asimismo, se podrá mostrar dicho valor en diferentes periodos de tiempo con el fin de comparar su variación.

- **Métricas de compromiso**

- **Crecimiento de publicaciones:** consiste en comparar cuánto ha crecido el número de publicaciones en un periodo. En este caso, no se darán fechas de inicio y fin completas sino un año de inicio y otro de fin y también se calculará mediante la Ecuación 4.1 sustituyendo el número de usuarios por el de publicaciones.
- **Crecimiento de coautores:** para cada autor se hallará, mediante la Ecuación 4.1, el crecimiento de sus relaciones de coautoría entre dos años dados por el usuario.
- **Crecimiento de las citas de una publicación:** se calculará el incremento, de nuevo con la Ecuación 4.1, en el número de citas que recibe una publicación en el periodo establecido por el usuario de la interfaz.
- **Número de veces que una publicación ha sido citada:** pudiendo calcularse el número total desde que se publicó o en un periodo determinado por el usuario de la aplicación.
- **Número de veces que han sido citadas las publicaciones de un determinado autor:** para cada autor se calculará el número de citas que han recibido sus publicaciones en un periodo de tiempo establecido por el usuario de la aplicación.
- **Fortaleza de los enlaces:** una vez establecida una relación de coautoría o amistad entre dos autores, se contabilizará el ratio de las publicaciones que han realizado juntos respecto a la suma del total de artículos que han publicado, pudiendo incluir nuevamente límite temporal, y haciendo uso de la Ecuación 4.3.
- **Fidelidad de coautores:** tras el establecimiento de una relación de coautoría entre dos autores, se calculará cómo de fieles son dichos autores entre sí. Para ello, se calculará el ratio de citas realizado por uno de ellos a

artículos publicados por el otro autor implicado en la coautoría respecto al total de citas realizadas por el primer autor, para lo cual se hará uso de la Ecuación 4.4. Al igual que en otros indicadores, se puede incluir un límite temporal.

- **Métricas de influencia**

- **Proximidad:** de cada autor se calculará qué porcentaje de las publicaciones que citan sus obras se encuentran a diversas proximidades (por ejemplo, la proximidad 1 corresponderá a aquellas citas cuyos autores hayan trabajado previamente con ese autor, la proximidad 2 a autores que hayan trabajado con autores con los que tengan relación de coautoría, etc.). De esta manera, se determinará qué influencia tiene dicho autor. En este caso, también habrá que tener en cuenta las auto-citas, a las que se asignará proximidad 0.

Nuevamente, y de la misma manera que ocurría con los conjuntos de datos anteriores, con DBLP también es posible obtener datos referentes al año en el que se publicaron más artículos, qué autores son los que más publican o quiénes son citados un mayor número de veces.

## 5 Diseño de la solución

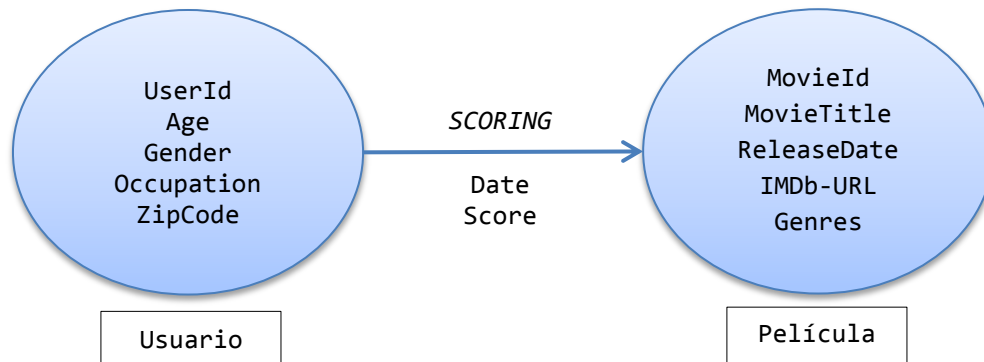
Este capítulo tiene como objetivo mostrar, en primer lugar, los diseños de las bases de datos en grafo utilizadas para trabajar con cada conjunto de datos y, posteriormente, describir el almacén de datos diseñado así como cada una de las tablas que lo constituyen. Finalmente, se explica el proceso realizado para trasladar los datos contenidos en las bases de datos orientadas a grafos al almacén de datos.

### 5.1 Bases de datos en grafo

Este apartado se centra en la descripción de las bases de datos en grafo creadas para cada uno de los conjuntos de datos empleados, utilizando para ello el gestor Neo4J, así como los cálculos realizados sobre ellas.

#### 5.1.1 Base de datos del conjunto de datos MovieLens

Como se mencionó en el apartado 4.2.1, este conjunto contiene información procedente del sitio web MovieLens, orientado a que los usuarios (de los que se conocen sus datos demográficos) realicen votaciones sobre películas, las cuales aparecen junto a su título, su fecha de lanzamiento, su dirección en el sitio web IMDb y los géneros a los que pertenecen. Conociendo esta información, y teniendo también en cuenta la fecha, hora y valoración de cada votación realizada, el diseño de la base de datos en grafo es el mostrado en la Ilustración 5.1.



**Ilustración 5.1** Grafo del conjunto de datos MovieLens

Tanto los usuarios como las películas se representan en el grafo mediante nodos en los que se recogen sus propiedades. Cabe destacar que para identificar cada nodo se utilizará, en los nodos que representan usuarios, la propiedad *UserId* y, en los nodos que representan películas, el valor de *MovieId*. Para representar las valoraciones que los usuarios realizan sobre las películas se utilizan aristas que relacionan al usuario que vota con la película que es votada. Esta relación tiene, además, dos propiedades: la fecha y hora en la que se realiza la votación y la valoración que otorga el usuario. En este caso, las votaciones no disponen de ninguna propiedad que las identifique, por lo que será el propio gestor el encargado de otorgarles un identificador único. Otro factor a tener en cuenta es el tipo al que pertenece la arista: aunque en este conjunto no sería necesario debido a que es la única relación que puede darse entre nodos, el gestor obliga a que todas las aristas pertenezcan a un tipo al que, en este caso, se ha llamado *SCORING*.

Teniendo presente el diseño necesario, se procedió a la escritura de diversas clases y métodos Java con el fin de cargar la información de MovieLens (almacenada en ficheros de texto planos) en la base de datos orientada a grafos, para lo que se hizo uso de una de las características que ofrece Neo4J a la hora de cargar datos por primera vez: la carga masiva, con la que se evitan transacciones mejorando así la velocidad de inserción.

Una vez cargados los datos, el grafo resultante constaba de 2625 nodos (de los cuales 943 correspondían a usuarios y 1682 a películas) y 100.000 aristas o relaciones, correspondientes a votaciones. Mediante el uso de este grafo se facilitaban operaciones tales como la obtención del número de votos recibidos por una película (sumando el número total de aristas a las que está conectado el nodo que la representa) o el número de votos emitidos por un usuario (nuevamente, calculando el número de aristas a las que está conectado el nodo usuario). Para llevar a cabo estos cálculos, se utilizaron métodos propios de Neo4J (por ejemplo, dado un nodo, es posible obtener todas las aristas de un determinado tipo a las que está conectado). Estos métodos utilizan a su vez de estructuras típicas de Java, como los iteradores.

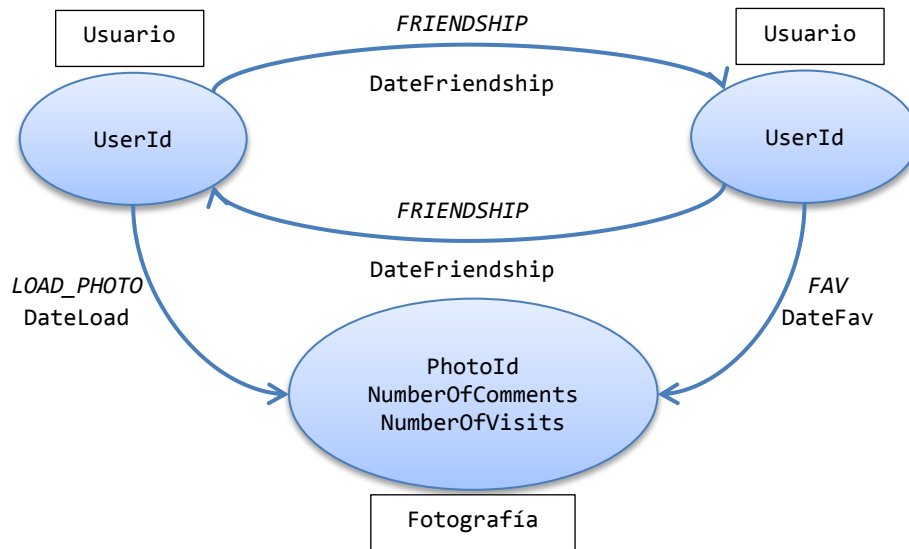
Haciendo uso de estas facilidades se llevó a cabo el recorrido de todos los nodos y aristas del grafo con el fin de obtener información relevante para la solución final. Algunos de estos datos calculados fueron, por ejemplo, el número de votos emitido por cada usuario o el número de votos que recibió una película. Toda la información extraída se almacenó en varios ficheros de texto organizados de forma que la lectura posterior fuese clara. Puesto que en el almacén (aspecto que se explicará en profundidad en el apartado 5.2) la dimensión encargada de almacenar fechas contiene datos cuyo cálculo resultaría complejo mediante el uso de SQL (como, por ejemplo, saber si una fecha pertenece a las vacaciones de Pascua) también se almacenaron en un fichero las diferentes fechas presentes en el grafo (fechas de estreno de películas o de votación de éstas) junto con información referente a ellas.

### 5.1.2 Base de datos del conjunto de datos Flickr

Un grafo adecuado para este conjunto de datos debe ser capaz de recoger usuarios, las fotografías subidas por estos, las relaciones de amistad que se establecen entre ellos y las votaciones que se realizan sobre las fotografías. Aunque de cada usuario tan sólo se conoce su identificador, de cada fotografía se dispone, además de su identificador, de la fecha y hora en la que se subió a la red social, del número de visitas y del número de comentarios recibidos. Asimismo, por cada relación de amistad se proporciona, además de los usuarios implicados, la fecha en la que se produjo y, por cada valoración de un usuario sobre una fotografía (limitándose dicha votación a marcarla como favorita), se tiene su fecha y su hora. Con esta información, una estructura adecuada de grafo para su almacenamiento es la mostrada en la Ilustración 5.2.

En ella, los usuarios y las fotografías aparecen representados mediante nodos mientras que las relaciones de amistad, las votaciones y la carga de las fotografías se representan mediante aristas, las cuales pertenecen a diferentes tipos (*FRIENDSHIP*, *FAV* y *LOAD\_PHOTO*, respectivamente) y todas ellas tienen como propiedad la fecha y hora en la que se producen. Al igual que ocurría con MovieLens, los identificadores de los usuarios y de las fotografías se utilizan para identificar los nodos y, además, puesto que cada relación de amistad y de votación consta también de un identificador, éste se emplea para

diferenciar la arista. No ocurre lo mismo con las relaciones de carga de fotografías, en las cuales el propio gestor se encarga de otorgarles una clave que las identifique.



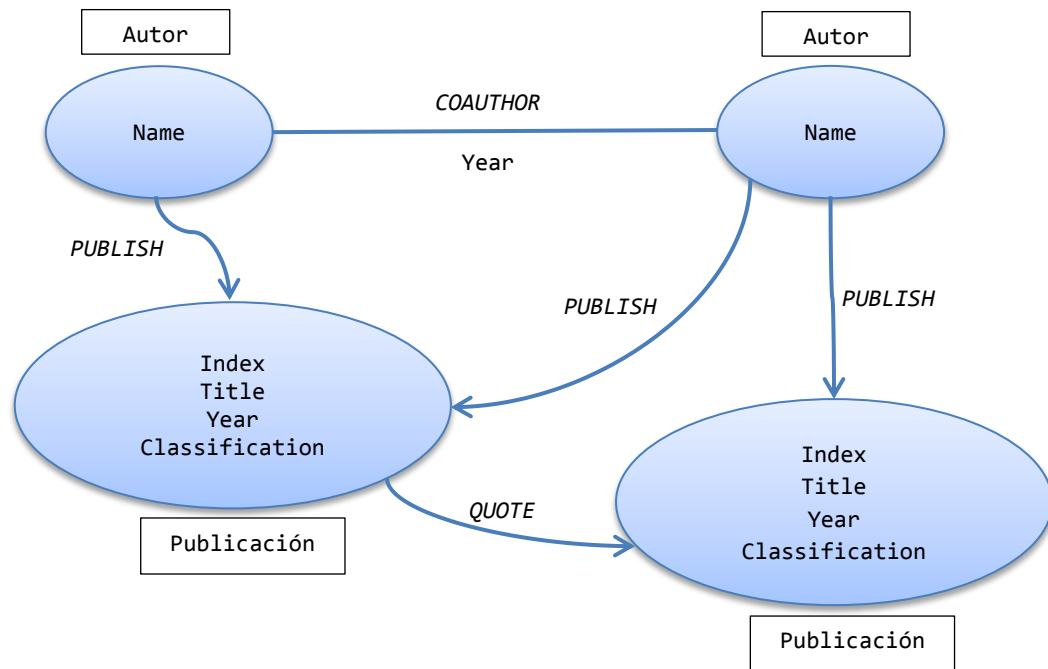
**Ilustración 5.2** Grafo del conjunto de datos Flickr

El proceso de carga desde los ficheros de origen y la base de datos en grafo se realizó de igual manera que en el conjunto MovieLens pero adaptando las operaciones de carga a las características del conjunto. El resultado final fue un grafo formado por 26.558.755 nodos y 25.707.885 aristas.

Gracias a este grafo se pudieron llevar a cabo operaciones propias de esta estructura y de utilidad para el cálculo de indicadores, entre los que destacan el cálculo de la proximidad entre dos usuarios (tanto en las relaciones de amistad como en la votación de fotografías y para lo que fue fundamental el uso de clases pertenecientes a Neo4J y que implementan algoritmos para trabajar con grafos) y la reciprocidad en las amistades (como se observa en la Ilustración 5.2, las relaciones de amistad son dirigidas puesto que si un usuario se hace amigo o seguidor de otro usuario no significa que la amistad se dé también en la dirección inversa). Para el cálculo de la reciprocidad se utilizó una consulta escrita en Cypher, lenguaje mencionado en el apartado 2.3.5, y con el que se consiguió simplificar este cálculo. Al igual que con MovieLens, se recorrió todo el grafo para obtener información referente a los usuarios, fotografías y relaciones acompañados de los cálculos mencionados y se almacenaron nuevamente en ficheros de texto. Otro paso en común con MovieLens fue el almacenamiento de fechas: por cada fecha presente en el grafo se realizaron las operaciones necesarias para extraer la información relevante y sus resultados se almacenaron en un fichero destinado a tal fin.

### 5.1.3 Base de datos del conjunto de datos DBLP

Para almacenar correctamente este conjunto de datos, el grafo debe tener una estructura que permita almacenar publicaciones de las que se conoce su título, sus autores, el año en el que se publicaron, las publicaciones a las que hace referencia, su clasificación y su índice dentro del conjunto. Con todo ello, una estructura de grafo adecuada es la mostrada en la Ilustración 5.3.

**Ilustración 5.3** Grafo del conjunto de datos DBLP

Los autores y las publicaciones se representan mediante nodos, los cuales recogen la información referente a los elementos que representan. Tanto para los nodos que representan autores como los que representan publicaciones se optó por dejar que el gestor Neo4J les otorgara un identificador puesto que los datos en origen no les otorgaban un atributo que los identificase. Para representar la relación entre las publicaciones y sus autores se utiliza la relación *PUBLISH* y, dado que una publicación puede estar escrita por varios autores, esta relación de coautoría debe recogerse, lo cual se lleva a cabo mediante las aristas de tipo *COAUTHOR*, que relacionan dos autores que han escrito juntos una publicación y la fecha que se otorga a esa “amistad” es el año en el que escribieron juntos por primera vez, de forma que dichos autores quedan relacionados desde su primera publicación en común. Otro aspecto representado mediante aristas son las referencias entre publicaciones, para las cuales se utilizan las aristas de tipo *QUOTE*. Tras la carga de los datos de origen, el grafo final consta de 2.393.769 nodos y 6.040.064 aristas.

Nuevamente, se procedió al recorrido del grafo y a la realización de los cálculos necesarios para su almacenamiento en ficheros. Algunos de los cálculos realizados que se apoyaban en la estructura del grafo fueron la obtención, por cada autor, de su número de publicaciones, de su número de coautores o del número de publicaciones que ha citado en sus publicaciones. En todos los casos se contabilizaron las aristas de cada tipo relacionadas con el nodo encargado de representar al autor y a las publicaciones de éste. Otro cálculo que se llevó a cabo fue la obtención de las proximidades, tanto en las relaciones de coautoría (calculando la proximidad entre autores antes de escribir juntos siguiendo las relaciones de coautoría que les separaban) como en las de cita a otras publicaciones, donde se tomó como proximidad el menor valor obtenido al calcular la proximidad de todos los autores que citan con todos los autores de la publicación citada.



## 5.2 Almacén de datos

A la hora de realizar el diseño del almacén de datos en el que se apoya la solución, se tuvieron en cuenta, principalmente, dos factores. El primero de ellos fue que dicho almacén de datos fuera capaz de responder a las diferentes métricas definidas en el apartado 4.1 y, el segundo de ellos, que fuera fácilmente extensible a cualquier conjunto de datos procedente una red social.

Esta generalización supuso llevar a cabo una práctica que, si bien no es recomendable en los diseños de almacenes de datos, en este caso sí fue necesaria: con el fin de adaptarse a las posibles diferencias entre conjuntos de datos, fue necesario que ciertas columnas admitieran valores nulos. Por ejemplo, mientras que en algunos conjuntos de datos los productos están relacionados con los usuarios que los publican, en otros será la propia red social la que los publique, por lo que no se dispondrá de ese valor. Con todo ello, el almacén de datos diseñado se muestra en la Ilustración A.1.

A continuación se describen todas las tablas que constituyen el almacén de datos y cómo se relacionan entre ellas.

### 5.2.1 Tabla *USER\_DIMENSION*

El objetivo de esta tabla es el almacenamiento de los usuarios junto a algunos de sus datos demográficos, tales como su nombre, sus apellidos o su sexo, o la fecha y hora en la que se dieron de alta en la red social (*RegisterDateKey* y *RegisterHourKey*). Dispone, además, de una columna para identificar a cada usuario dentro del almacén de datos (*UserKey*) y que será la clave primaria de la tabla y de otra columna en la que se almacenará el identificador del usuario en los datos de origen, necesario para la tarea de sincronización. Asimismo, y contemplando el hecho de que puedan existir grupos de usuarios, se incluye la columna *IsGroup*, en la que se indicará si esa fila de la tabla representa a un grupo o a un usuario individual. En el caso de que sea un grupo de usuarios, tan sólo se tendrá valor en la columna *UserKey* siendo nulos los valores del resto de columnas.

### 5.2.2 Tabla *GROUP\_USER*

Esta tabla recoge los usuarios que pertenecen a un grupo, por ejemplo cuando varios usuarios publican juntos un artículo científico. Consta de dos columnas: una se encarga de identificar al grupo y la otra recoge, en cada fila, cada uno de los usuarios que constituyen ese grupo.

### 5.2.3 Tabla *USERSNAPSHOT*

Con el fin de facilitar las consultas a realizar, se dispone de esta tabla de hechos de tipo *snapshot* en la que se recogen datos de la actividad de un usuario en un periodo de tiempo dado, lo que se indica mediante dos columnas en las que se hace referencia a las fechas de inicio y fin de dicho periodo. Entre los valores que recoge esta tabla de hechos están, además de una referencia al usuario al que pertenece la información, datos referentes a su edad, ocupación o código postal en ese periodo (incluidos en esta tabla puesto que pueden variar en diferentes periodos de tiempo) así como información de su actividad en la red social tales como el número de productos publicados, número de votaciones realizadas, número de amigos, número de amigos de sus amigos o si fue líder durante ese periodo de tiempo (Palazuelos & Zorrilla, 2012). Asimismo, recoge una serie de medidas, como el

número de días, semanas o meses transcurridos en el periodo de tiempo que corresponda, cuyo fin es facilitar futuras consultas.

#### 5.2.4 Tabla *USER\_SESSION*

El cometido de esta tabla es recoger información de las sesiones iniciadas por el usuario en la red, como la fecha y hora en la que inició la sesión, la fecha y hora en la que cerró dicha sesión, el tiempo que permaneció y la actividad realizada en ella, como por ejemplo el número de votos, mensajes o invitaciones enviados o el número de productos o comentarios publicados.

#### 5.2.5 Tabla *AGE*

Esta tabla almacena diferentes rangos de edad que pueden tener los usuarios, para lo que se dispone de columnas donde se indica el valor mínimo y máximo que comprende cada rango.

#### 5.2.6 Tabla *AREA*

Su objetivo es almacenar diferentes códigos postales que indiquen en qué lugar residen los usuarios.

#### 5.2.7 Tabla *OCUPATION*

Se encarga de almacenar los estudios u ocupaciones que tienen los usuarios.

#### 5.2.8 Tabla *PRODUCT\_DIMENSION*

Esta tabla representa los diferentes productos o contenidos que se publican en la red social (fotografías, películas, etc.). Entre las columnas que la forman están, además de los identificadores, la fecha y hora en el que fue publicado, el usuario responsable de ello y una descripción (si, por ejemplo, se representan películas esta columna se utiliza para almacenar el título de éstas).

#### 5.2.9 Tabla *PRODUCTSNAPSHOT*

De igual forma que se dispone de la tabla *USERSNAPSHOT* para los usuarios, también se tiene una tabla de hechos con el mismo cometido pero, esta vez, dirigida a los productos. Las columnas que forman esta tabla se encargan de representar valores relacionados con el producto, como el número de votos, comentarios, visitas y valoración media recibidos entre las fechas de inicio y fin indicadas para ese *snapshot* así como medidas que faciliten futuras consultas, al igual que ocurría en la tabla *USERSNAPSHOT*.

#### 5.2.10 Tabla *CLASSIFICATION*

Puesto que ciertos productos, como películas o libros, pueden pertenecer a varios géneros, se dispone de esta tabla para almacenar cada uno de estos posibles géneros o clasificaciones.

#### 5.2.11 Tabla *GENRE*

Dado que es habitual que un producto pertenezca a varios géneros de los almacenados en la tabla *CLASSIFICATION*, la tabla *GENRE* será necesaria a la hora de relacionar cada producto con los diferentes géneros a los que pertenece.

#### 5.2.12 Tabla *DATE\_DIMENSION*

Esta tabla se encarga de representar la dimensión temporal, de vital importancia en cualquier almacén de datos y que contiene atributos referentes a la fecha (Kimbal & Ross,

2002), como el día del año, si es día festivo, si es el último fin de semana del mes o si pertenece a un periodo vacacional. Puesto que es posible que en algunos casos no se disponga de la fecha completa, la mayor parte de sus columnas deberán admitir valores nulos.

#### 5.2.13 Tabla *HOURL\_DIMENSION*

Puesto que se tendrá información referente al momento del día en el que se realizan ciertas acciones en la red social, es necesario disponer de una dimensión en la que se recoja la hora y el minuto del día en el que se produjo dicha acción.

#### 5.2.14 Tabla *FRIENDSHIP*

Esta tabla de hechos transaccional recoge las relaciones que se dan entre usuarios, para lo que se dispone de columnas que identifiquen a dichos usuarios así como la fecha y hora en la que se produjo dicha amistad. Asimismo, se dispondrá de dos columnas que almacenarán medidas referentes a cada relación: la proximidad entre los usuarios implicados antes de que existiera la relación de amistad y si dicha amistad es recíproca.

#### 5.2.15 Tabla *SCORE*

El objetivo de esta tabla de hechos transaccional es almacenar las votaciones o referencias que los usuarios realizan sobre productos publicados en la red. De cada votación se podrá almacenar, además del usuario y el producto implicados, la fecha y la hora en la que se produjo dicha votación, la valoración otorgada y la proximidad existente entre el usuario que vota y el usuario propietario del producto.

#### 5.2.16 Tabla *RANGE*

Puesto que en función de la red social varían las valoraciones mínimas y máximas que los usuarios pueden otorgar a los productos, en esta tabla se recogen los rangos de puntuaciones que se pueden otorgar utilizando para ello columnas que representen los valores mínimo y máximo.

### 5.3 Proceso ETL

En este apartado se describe el proceso llevado a cabo para trasladar la información extraída de cada uno de los grafos hasta el almacén de datos diseñado.

Como se mencionó, de cada grafo se extrajo información que se almacenó en ficheros haciendo uso de métodos y clases que proporciona el gestor Neo4J para trabajar sobre grafos. Una vez almacenada en ficheros la información relevante de cada conjunto de datos, el siguiente paso fue el traslado de esos datos al almacén de datos.

Debido a la gran cantidad de dependencias existente en el almacén de datos y en vista a futuras actualizaciones, insertar directamente los ficheros extraídos en el almacén suponía, además de una pérdida de rendimiento debido al volumen de la información, una gran probabilidad de cometer errores por las dependencias existentes y dificultades en futuras actualizaciones. Para solventar estos problemas, se hizo uso de un *data staging* para cada conjunto de datos.

Los *data staging*, mencionados en el capítulo 3, están muy presentes en las soluciones Business Intelligence y se definen como una base de datos relacional intermedia entre las fuentes de información y el almacén de datos cuya finalidad es

facilitar la carga de datos en éste así como futuras actualizaciones. En el caso de este proyecto, se ha diseñado un *data staging* relacional para cada conjunto de datos de forma que actúe como paso intermedio entre los ficheros extraídos de los grafos y el almacén diseñado. Una vez que la información de los ficheros se encuentra almacenada en su *data staging* correspondiente, el traslado al almacén de datos se simplifica.

El *data staging* diseñado para el conjunto de datos MovieLens se muestra en la Ilustración 5.4 y permite recoger la información extraída del grafo de forma ordenada además de facilitar futuras actualizaciones (por ejemplo, con la presencia de las tablas *USER\_ACTIVITY* y *MOVIE\_ACTIVITY*, que recogen información referente a usuarios y películas en el periodo de tiempo en el que se haya recogido dicha información).

El *data staging* de Flickr, mostrado en la Ilustración 5.5, es similar al anterior pero añadiendo la tabla *FRIENDSHIP*. Dos de las columnas de esta tabla (*Proximity* y *Reciprocity*) recogen información que, sin el uso de una base de datos en grafo, hubieran resultado muy complejas o incluso imposibles de calcular.

Finalmente, el *data staging* de DBLP, representado en la Ilustración 5.6 también tiene características similares a los anteriores así como particularidades (por ejemplo, consta de la tabla *GROUP\_AUTHORS* cuyo fin es almacenar los grupos de escritores que publican juntos un artículo).

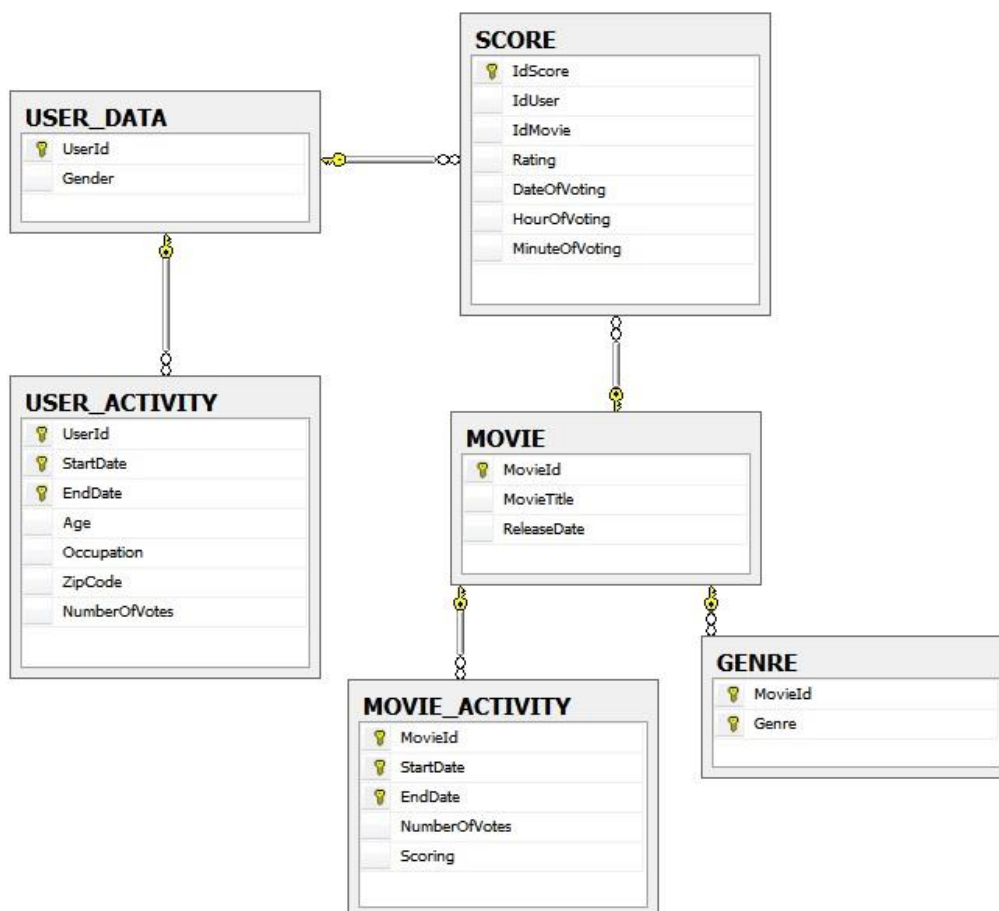


Ilustración 5.4 Data staging para el conjunto de datos MovieLens

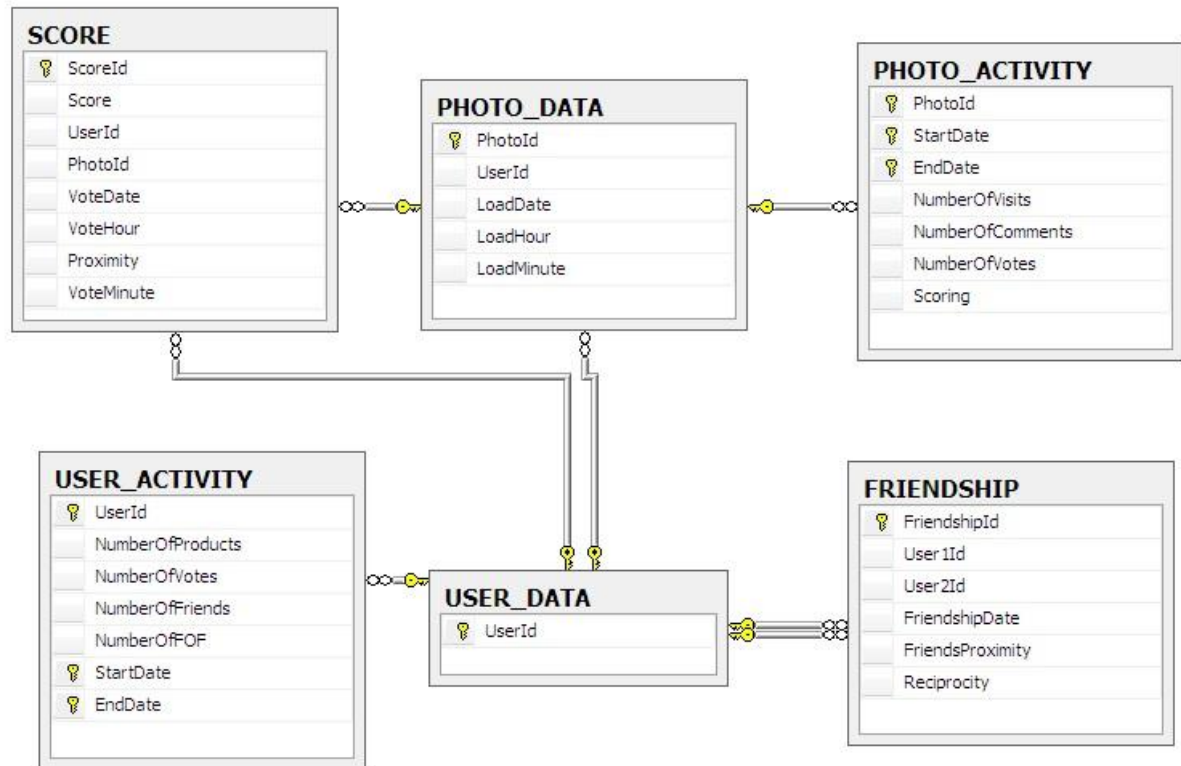


Ilustración 5.5 Data staging para el conjunto de datos Flickr

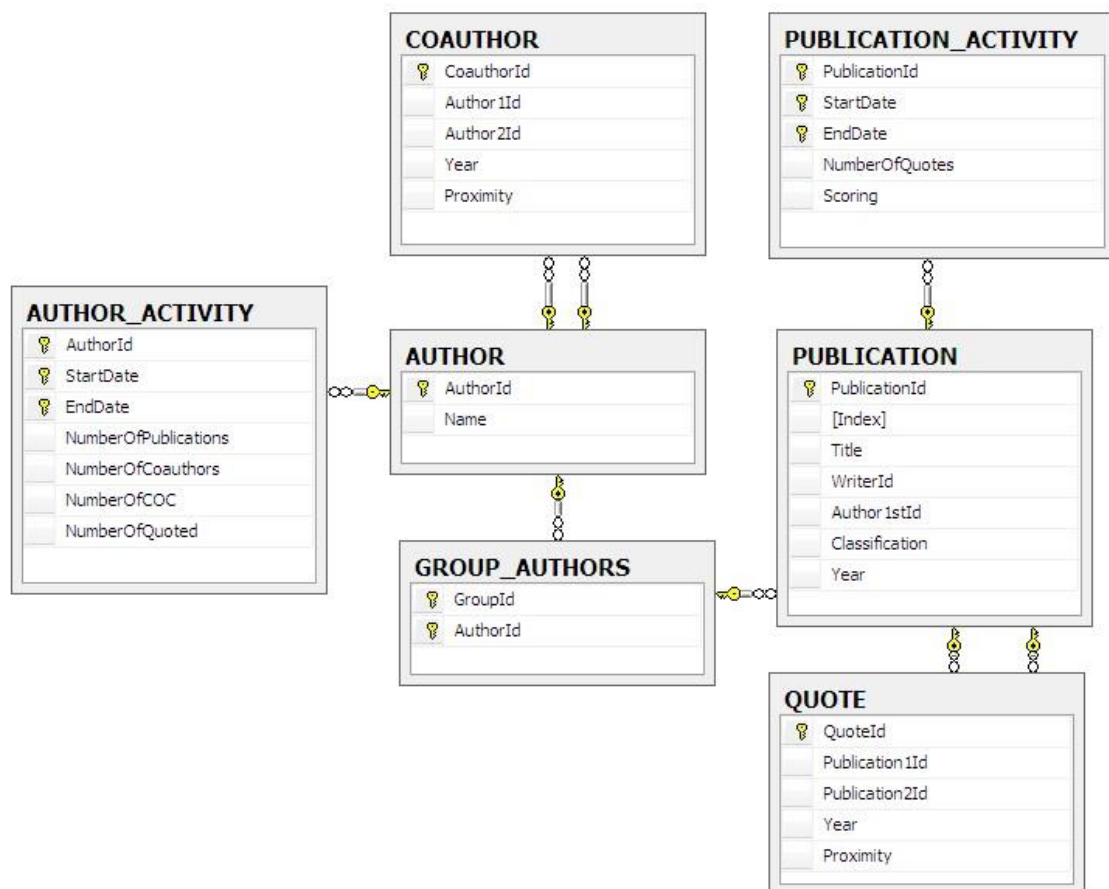
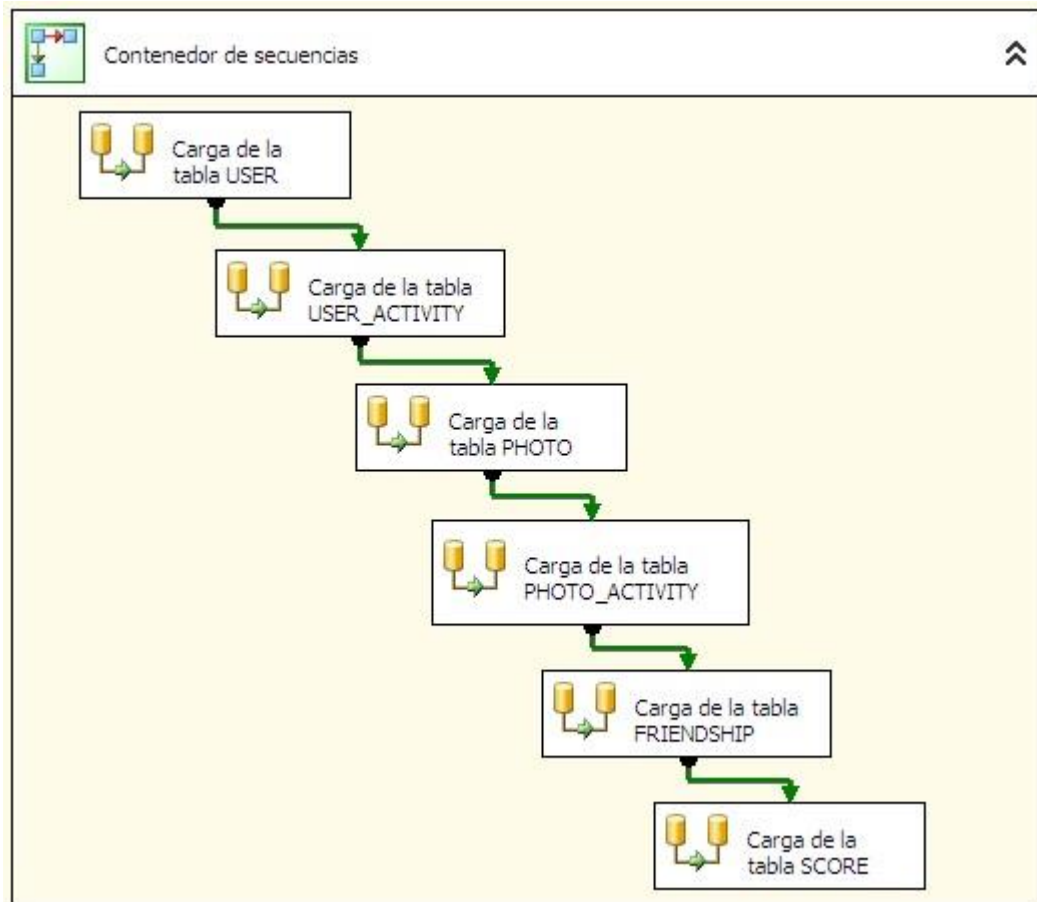


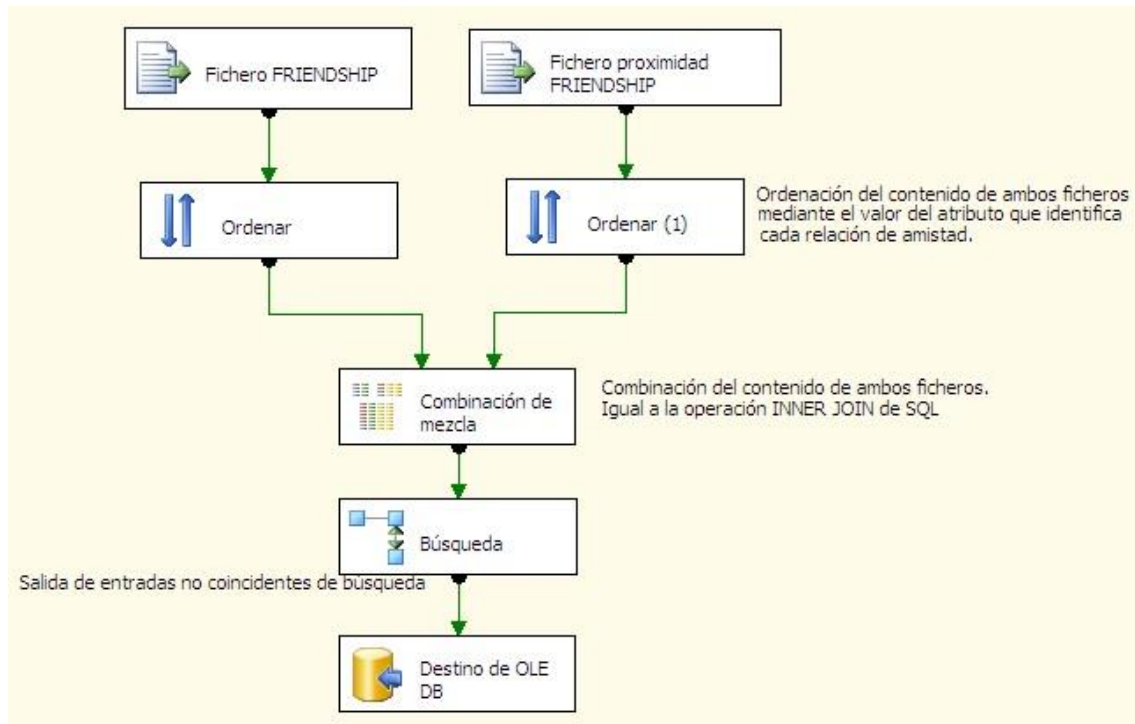
Ilustración 5.6 Data staging para el conjunto de datos DBLP

A la hora de insertar los datos extraídos de cada grafo en su *data staging* correspondiente se hizo uso de la plataforma SQL Server Integration Services (en adelante SSIS), una de las herramientas de SQL Server Business Intelligence Development Studio orientada al proceso ETL y que se describió en el apartado 3.3. Esta herramienta proporciona facilidades que simplifican la inserción de datos pudiendo hacerse la mayor parte de ellas de forma gráfica. En la Ilustración 5.7 se muestra una captura del proceso ETL diseñado mediante SSIS para la carga de datos en el *data staging* de Flickr, donde aparece la carga de las distintas tablas de forma secuencial con el fin de evitar problemas de referencias.



**Ilustración 5.7** Proceso de carga en el *data staging* de Flickr

Dentro de cada uno de estos flujos de datos se realiza la lectura de datos desde el fichero de texto, una búsqueda en el *data staging* para evitar datos duplicados y, finalmente, la inserción en la tabla correspondiente. La carga de algunas tablas, como *FRIENDSHIP*, mostrada en detalle la Ilustración 5.8, tiene ciertas particularidades. En este caso, además de la lectura, la búsqueda y la carga, fue necesario unir información que se encontraba almacenada en dos ficheros diferentes.

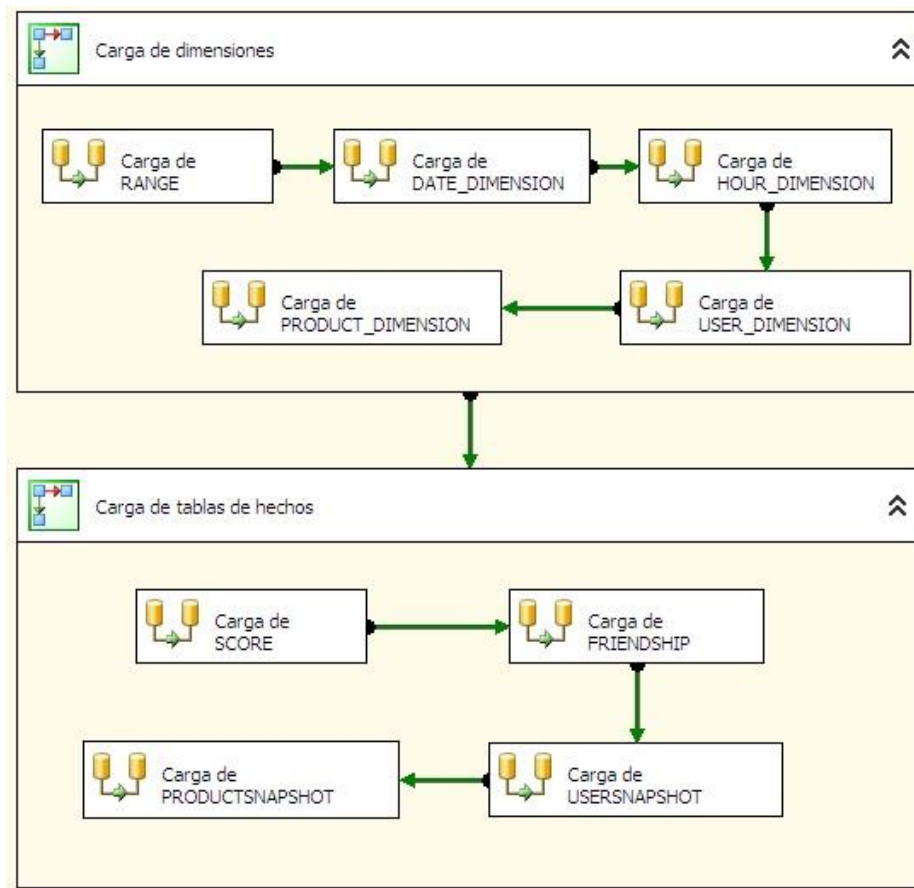


**Ilustración 5.8** Proceso de carga de la tabla *FRIENDSHIP*

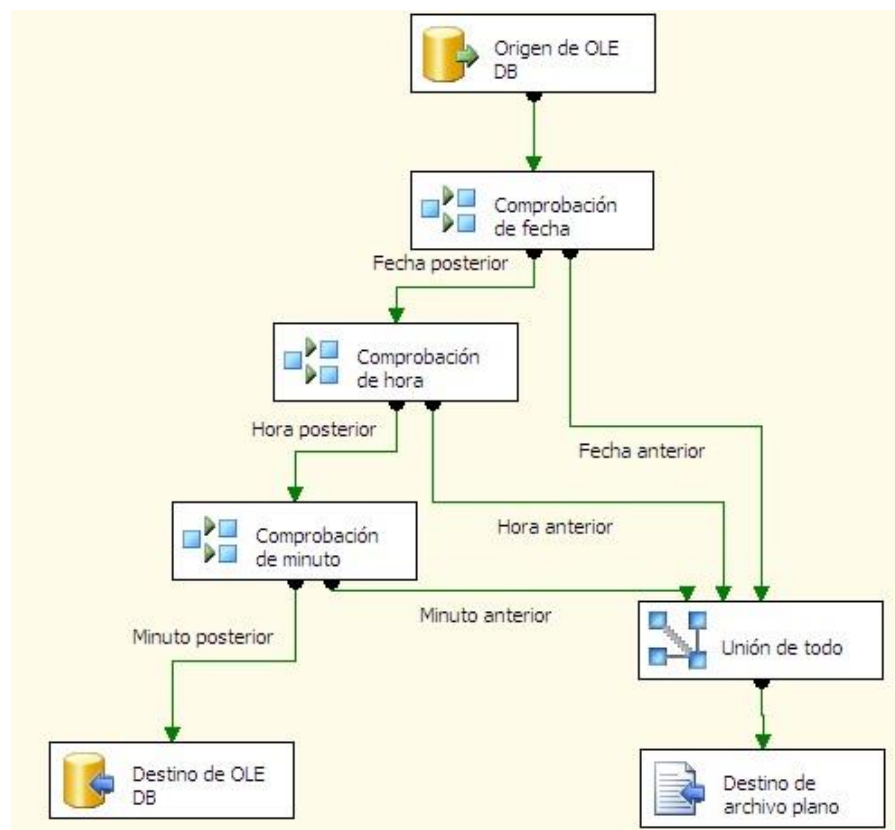
Una vez que los datos de cada conjunto de datos se encuentran cargados en su *data staging*, el siguiente paso es la carga de dichos datos en el almacén de datos. Para ello, se hizo uso nuevamente de la herramienta SSIS y de las facilidades que ofrece. En la Ilustración 5.9 se muestra una captura del flujo de datos diseñado para la carga de los datos procedentes del conjunto de datos Flickr.

En este paso, el esquema seguido es similar al anterior que se vio en la Ilustración 5.8. Un factor a tener en cuenta es la presencia de datos “sucios” como, por ejemplo, que la información correspondiente al día y hora de votación de una fotografía refleje un momento anterior a la carga de dicha fotografía<sup>31</sup>. Para localizar este tipo de errores, fue necesario incluir una tarea que detectara estos datos y los descartara antes de incluirlos en el almacén de datos, proceso que se muestra en la Ilustración 5.10, donde también se incluyó una tarea que se encargara de almacenar los datos erróneos en un fichero de texto.

<sup>31</sup> Por tratarse de conjuntos de datos públicos es probable la presencia de datos que, en detalle, contienen errores.



**Ilustración 5.9** Proceso de carga de los datos del *data staging* de Flickr en el almacén de datos



**Ilustración 5.10** Eliminación de las votaciones de Flickr con datos erróneos de fecha y hora



## 6 Generación de informes

En este capítulo se describe la interfaz a través de la cual el usuario puede acceder a los informes diseñados para representar las métricas definidas en el apartado 4.1 aplicadas sobre los conjuntos de datos cargados en el almacén de datos. Asimismo, se describen las medidas de seguridad implementadas para el acceso a los informes y las acciones realizadas para mejorar rendimiento de las consultas.

### 6.1 Herramienta Reporting Services

Como se mencionó en el apartado 3.3, SQL Server Business Intelligence Development Studio proporciona herramientas para facilitar el trabajo en soluciones Business Intelligence. En el caso de los informes, la herramienta que permite su definición y explotación es Reporting Services.

Esta herramienta permite conectar, entre otros, con bases de datos relacionales y con cubos OLAP de forma que se puedan realizar consultas sobre ellos y mostrar la información procedente de esas consultas con distintos formatos. En este caso, la mayor parte de las consultas implementadas para responder a las métricas se han realizado contra el almacén de datos pero aquellas en las que, debido a la complejidad de las consultas y a la gran cantidad de datos almacenados, se observaba una importante pérdida de rendimiento se optó por la construcción de cubos OLAP.

Otra característica importante de esta herramienta es la posibilidad de visualizar los informes creados a través de un servidor web integrado en la propia herramienta. Para ello, basta con implementar los informes creados para que sea posible su visualización conectándose al servidor a través de un explorador web. En el servidor dichos informes pueden imprimirse y exportarse a diferentes formatos como PDF, Excel o Word.

Este servidor web permitirá, además, la creación de carpetas en las que se podrán clasificar los informes en base a diferentes criterios con los que se conseguirá que el usuario de la interfaz pueda acceder a ellos de forma más rápida e intuitiva.

### 6.2 Descripción de informes

El objetivo de este apartado es la descripción de los informes creados y la visualización de los mismos en el servidor web ofrecido por Reporting Services. Iniciando un explorador de internet y conectándose a la dirección <http://localhost/Reports><sup>32</sup>, el servidor solicitará al usuario que introduzca su nombre y contraseña a través de un diálogo para acceder a la interfaz mostrada en la Ilustración 6.1.

En el caso de esta interfaz, el acceso a los informes se realiza a través de carpetas. Estas carpetas muestran los informes en función del conjunto de datos sobre el que trabajan (Flickr, MovieLens o DBLP) o de las métricas a las que responden (de alcanzabilidad, de compromiso o de influencia), como aparece reflejado en la Ilustración 6.2.

---

<sup>32</sup> Esta dirección puede cambiarse a través de la interfaz web en función de las necesidades o preferencias del usuario.

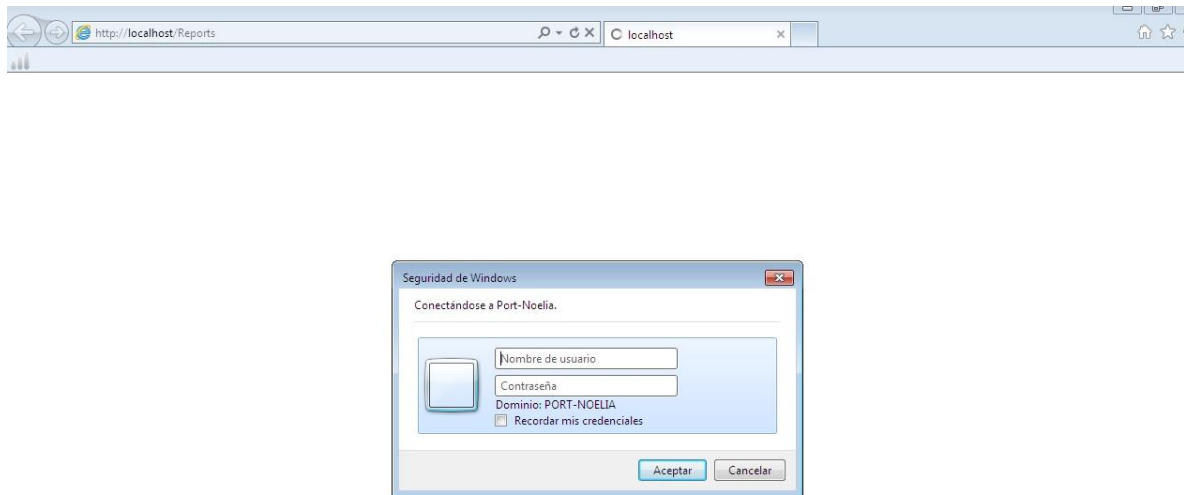


Ilustración 6.1 Identificación de usuario en la interfaz web

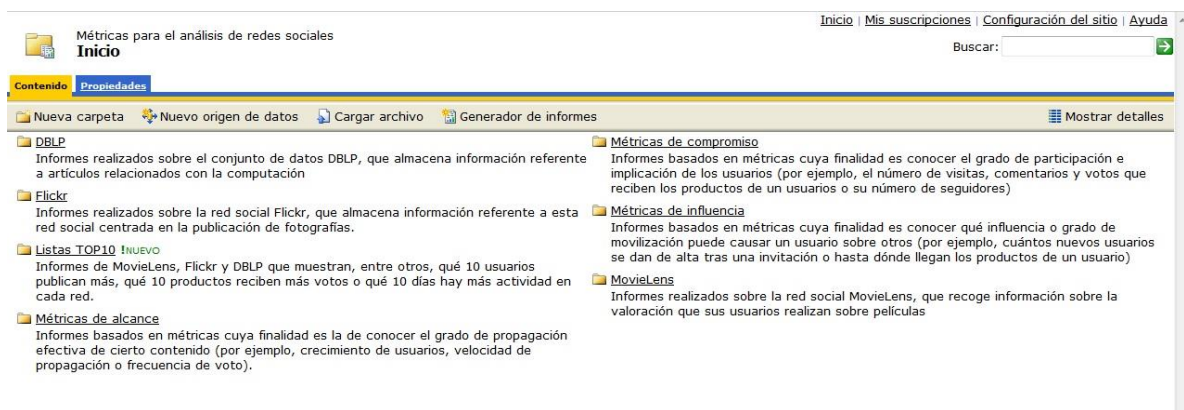


Ilustración 6.2 Formulario de inicio de la interfaz

Dentro de dichas carpetas se encuentran, respectivamente, los informes creados para cada conjunto de datos o para visualizar las métricas que correspondan. El primer caso puede verse en la Ilustración 6.3, donde se muestran los informes creados para el conjunto de datos Flickr. El segundo caso aparece en la Ilustración 6.4, correspondiente a la carpeta de métricas de alcanzabilidad y que engloba, independientemente del conjunto de datos al que correspondan, todos los informes que dan respuesta a métricas de ese tipo.

A la hora de escribir la consulta SQL para generar cada informe, fue necesario tener en cuenta las características del conjunto de datos al que correspondería dicho informe, es decir, se seleccionaron sólo aquellos datos que correspondían a dicho conjunto. Para ello, las consultas escritas contienen ciertos criterios de filtrado de datos: por ejemplo, para Flickr se seleccionaron tan sólo aquellos usuarios de los que no se tenía ni su nombre ni su sexo y, para MovieLens, aquellos productos que tuvieran una descripción pero ningún usuario propietario. Gracias a estas diferencias entre los datos, no fue necesario incluir nuevas columnas en las tablas que indicaran, para cada fila, a qué conjunto de datos correspondía.

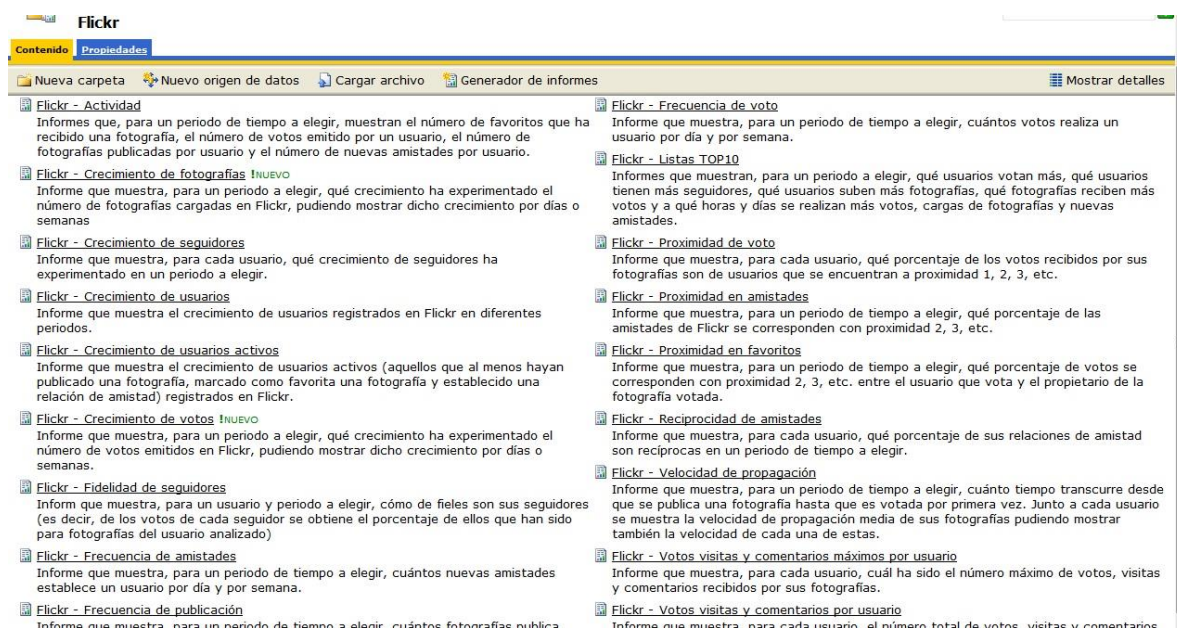


Ilustración 6.3 Carpeta de informes sobre Flickr

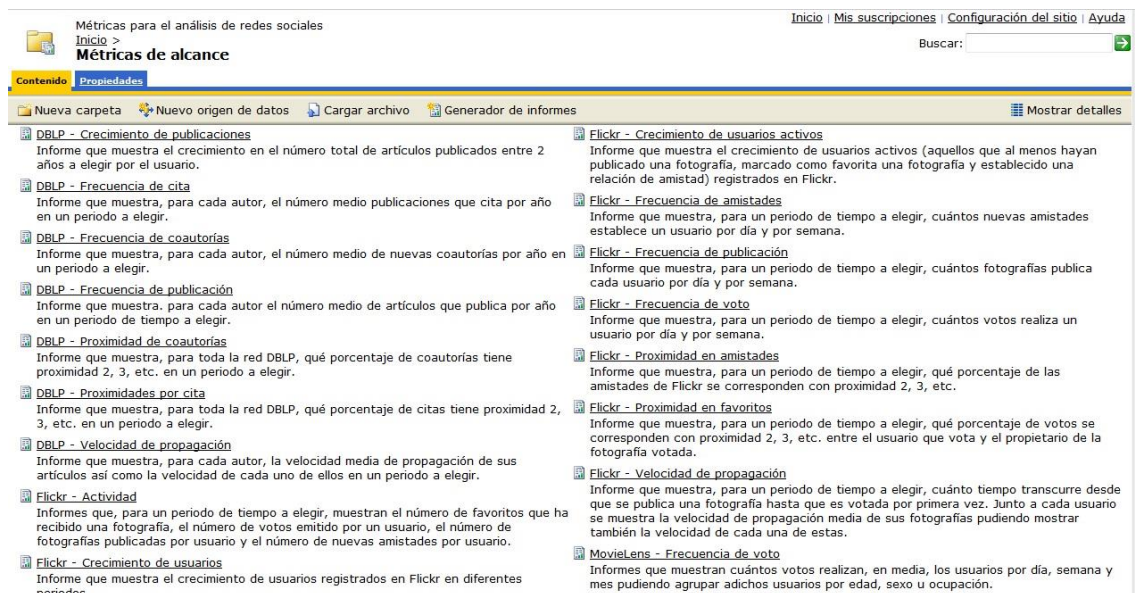


Ilustración 6.4 Carpeta de informes con métricas de alcance

En los siguientes apartados se describen algunos de los informes creados junto con una captura de los mismos.

### 6.2.1 Informe de crecimiento de usuarios

Este informe implementa la métrica de alcanzabilidad que contabiliza el crecimiento de usuarios que se ha producido en la red social en un periodo de fechas determinado<sup>33</sup>. Por los datos proporcionados en los conjuntos de datos, tan sólo se ha podido realizar dicho informe para Flickr, el cual se muestra en la Ilustración 6.5. Para facilitar la visualización, se ofrece la posibilidad de visualizar estos datos mediante una gráfica, como la mostrada en la Ilustración 6.6.

<sup>33</sup> En el caso del informe mostrado, correspondiente a Flickr, los periodos en los que se estudia el crecimiento son los correspondientes a los *snapshots* proporcionados.



Ilustración 6.5 Informe de crecimiento de usuarios

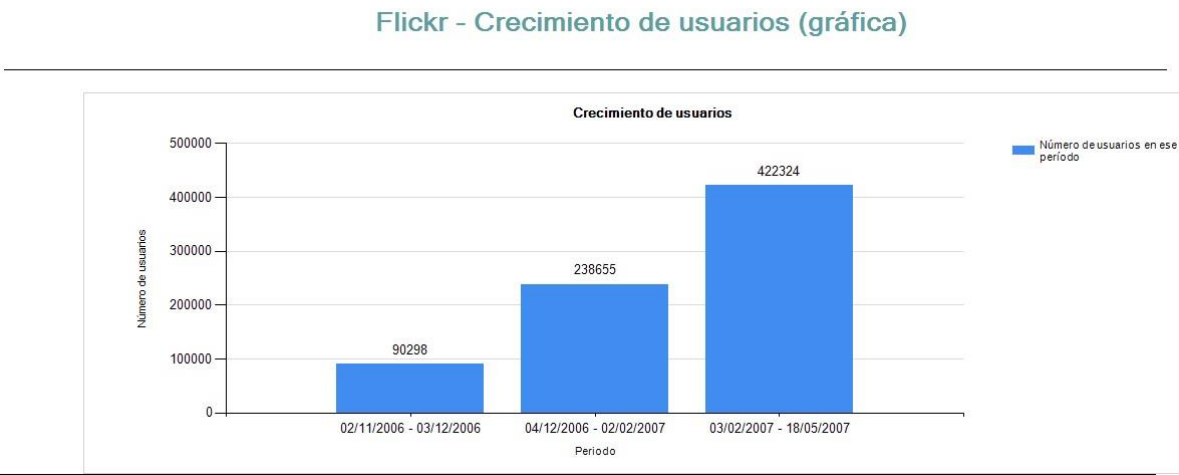


Ilustración 6.6 Gráfica del crecimiento de usuarios

6.2.2 Informe de crecimiento de usuarios activos

La estructura de este informe es igual al anterior pero, en este caso, se contabiliza tan sólo el crecimiento de usuarios activos.

6.2.3 Informe de velocidad de propagación

Esta métrica, perteneciente a las métricas de alcanzabilidad, se ha podido implementar para el conjunto de datos de Flickr y de DBLP. El primer caso, mostrado en la Ilustración 6.7, muestra el tiempo medio mínimo (días, horas y minutos) que transcurre desde que cada usuario publica sus fotografías hasta que son votadas por primera vez. Desplegando la fila correspondiente a cada usuario, se puede visualizar en detalle qué velocidad de propagación tiene cada fotografía. Además, se proporciona al usuario final la posibilidad de elegir mediante parámetros las fechas de inicio y fin para realizar la consulta así como la opción de poder ordenar los datos mostrados para conocer qué usuario tiene una menor o mayor velocidad de propagación en sus fotografías.

Fecha de inicio: 02/11/2006 Fecha de fin: 03/12/2006

1 de 1090 100% Buscar | Siguiente Seleccionar un formato Exportar

### Flickr - Velocidad de propagación

| Usuario | Fotografía | Diferencia en días | Diferencia en horas | Diferencia en minutos |
|---------|------------|--------------------|---------------------|-----------------------|
| 100     |            | 6.00               | 154.00              | 9242.67               |
|         | 14861650   | 0.00               | 0.00                | 1.00                  |
|         | 22555188   | 14.00              | 349.00              | 20944.00              |
|         | 15963051   | 4.00               | 113.00              | 6783.00               |
| 1000    |            | 8.42               | 208.50              | 12529.83              |
| 10000   |            | 2.00               | 57.75               | 3475.75               |
| 100001  |            | 9.00               | 216.00              | 12965.50              |
| 100002  |            | 0.00               | 0.00                | 0.00                  |
| 100003  |            | 6.50               | 162.60              | 9771.70               |
| 100008  |            | 2.60               | 68.40               | 4124.20               |
| 100010  |            | 0.00               | 0.00                | 0.00                  |
| 100013  |            | 4.33               | 120.67              | 7268.67               |
| 100020  |            | 10.00              | 240.00              | 14410.00              |

Ilustración 6.7 Informe de velocidad de propagación

#### 6.2.4 Informe de actividad en la red

Este informe muestra la actividad que se ha producido en la red entre dos fechas a elegir por el usuario. En el caso de la captura mostrada en la Ilustración 6.8, correspondiente al número de fotografías publicadas en Flickr, el usuario de la interfaz podrá elegir si desea conocer el número de votos recibidos por fotografía, el número de votos emitido por usuario, el número de fotografías publicadas por usuario o el número de nuevas amistades por usuario. Además, se ofrece la posibilidad de agrupar de mayor a menor los datos mostrados (en el caso de la captura mostrada, puede ordenarse según el número de fotografías publicado) así como el total de todos los usuarios.

Ver: Fotografías subidas Fecha de inicio: 02/11/2006

Fecha de fin: <Seleccione un valor>

Votos recibidos por fotografía

Votos emitidos por usuario

Fotografías subidas

Nuevas amistades

1 Buscar | Siguiente Seleccionar un formato

| Usuario | Número de fotografías |
|---------|-----------------------|
|         | 1988758               |
| 367253  | 2                     |
| 95499   | 2                     |
| 120086  | 2                     |
| 141326  | 3                     |
| 79309   | 24                    |
| 838329  | 1                     |
| 99492   | 5                     |
| 245638  | 2                     |

Ilustración 6.8 Informe de actividad

#### 6.2.5 Informe de frecuencia de publicación

Este informe recoge cuántos productos publica cada usuario, en media, al día, a la semana o al mes (en el caso de Flickr) o al año (en el caso de DBLP), el número total de productos publicados y qué porcentaje suponen sus publicaciones respecto al total de la red. El informe mostrado en la Ilustración 6.9, perteneciente a Flickr, permite además elegir en qué periodo de tiempo de los disponibles se desea estudiar dicha frecuencia. Asimismo, se



muestra el total de los productos publicados en la red, la frecuencia media de publicación y se ofrece la posibilidad de ordenar en función de varios criterios como el porcentaje sobre el total, el número total o las frecuencias.

Periodo a analizar: Del 02/11/2006 al 03/12/2006  
 <Seleccione un valor>  
 03/12/2006 al 03/12/2006  
 Del 04/12/2006 al 02/02/2007  
 Del 03/02/2007 al 18/05/2007

1 de 1 Buscar | Siguiente Seleccionar un formato Exportar

### Flickr - Frecuencia de publicación

| Usuario                  | Número de fotos | Porcentaje sobre el total | Fotos por día | Fotos por semana | Fotos por mes |
|--------------------------|-----------------|---------------------------|---------------|------------------|---------------|
| <b>Nº total de fotos</b> | <b>1749613</b>  | <b>Frecuencias medias</b> | <b>0,63</b>   | <b>3,88</b>      | <b>19,38</b>  |
| 70666                    | 19              | 0,0010%                   | 0,61          | 3,80             | 19,00         |
| 78304                    | 38              | 0,0019%                   | 1,23          | 7,60             | 38,00         |
| 2752                     | 18              | 0,0009%                   | 0,58          | 3,60             | 18,00         |
| 2522                     | 939             | 0,0472%                   | 30,29         | 187,80           | 939,00        |
| 120165                   | 1               | 0,0001%                   | 0,03          | 0,20             | 1,00          |
| 46810                    | 121             | 0,0061%                   | 3,90          | 24,20            | 121,00        |
| 578458                   | 8               | 0,0004%                   | 0,26          | 1,60             | 8,00          |
| 44233                    | 40              | 0,0020%                   | 1,29          | 8,00             | 40,00         |
| 98609                    | 2               | 0,0001%                   | 0,06          | 0,40             | 2,00          |
| 30781                    | 13              | 0,0007%                   | 0,42          | 2,60             | 13,00         |
| 25597                    | 5               | 0,0003%                   | 0,16          | 1,00             | 5,00          |
| 219722                   | 4               | 0,0002%                   | 0,13          | 0,80             | 4,00          |
| 132563                   | 1               | 0,0001%                   | 0,03          | 0,20             | 1,00          |

Ilustración 6.9 Informe de frecuencia de publicación

#### 6.2.6 Informe de frecuencia de voto

Este informe permite conocer cuántos votos realiza cada usuario, de media, en un día, una semana o un mes. Esta métrica, implementada para los tres conjuntos de datos empleados, se visualiza mediante informes como el mostrado en la Ilustración 6.10, correspondiente al conjunto de datos MovieLens y que proporciona, además, parámetros para agrupar estos datos según la edad, el sexo o la ocupación de los usuarios así como elegir una edad, sexo u ocupación específicos.

Agrupar en base a: Edad Edades: Todas  
 Ocupacion: Todas Sexo: Ambos

1 de 1 100% Buscar | Siguiente Seleccionar un formato Exportar

### MovieLens - Frecuencia de voto

| Ver gráfica de porcentaje de voto |         |                       |                           |               |                  |               |
|-----------------------------------|---------|-----------------------|---------------------------|---------------|------------------|---------------|
| Edad                              | Usuario | Número total de votos | Porcentaje sobre el total | Votos por día | Votos por semana | Votos por mes |
| De 0 a 17 años                    |         | 2491                  | 2,50 %                    | 11,00         | 83,00            | 355,00        |
| De 18 a 29 años                   |         | 45268                 | 45,38 %                   | 210,00        | 1508,00          | 6466,00       |
| De 30 a 59 años                   |         | 49421                 | 49,54 %                   | 229,00        | 1647,00          | 7060,00       |
| De 60 a 99 años                   |         | 2820                  | 2,83 %                    | 13,00         | 94,00            | 402,00        |
|                                   | 106     | 64                    | 0,06 %                    | 0,30          | 2,13             | 9,14          |
|                                   | 211     | 37                    | 0,04 %                    | 0,17          | 1,23             | 5,29          |
|                                   | 234     | 480                   | 0,48 %                    | 2,23          | 16,00            | 68,57         |
|                                   | 266     | 23                    | 0,02 %                    | 0,11          | 0,77             | 3,29          |
|                                   | 308     | 397                   | 0,40 %                    | 1,85          | 13,23            | 56,71         |
|                                   | 318     | 174                   | 0,17 %                    | 0,81          | 5,80             | 24,86         |

Ilustración 6.10 Informe de frecuencia de voto

En el caso de MovieLens, además de un informe en forma de tabla, se ofrece la posibilidad de que la información referente al porcentaje de votos emitido por cada grupo pueda visualizarse mediante gráficas, tal como se muestra en la Ilustración 6.11.

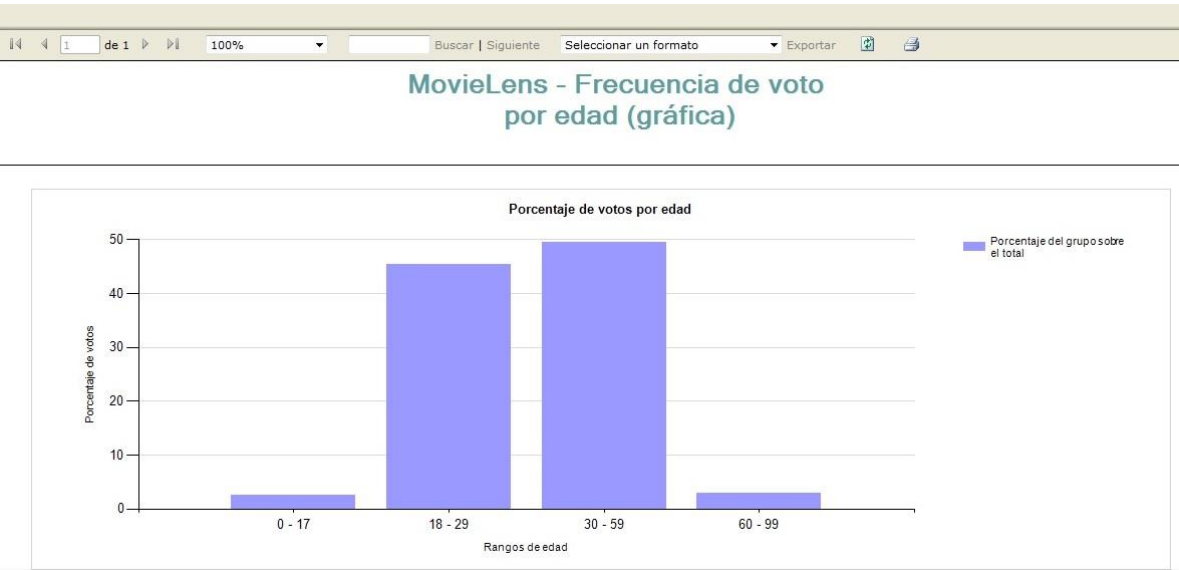


Ilustración 6.11 Gráfica de porcentaje de voto por rango de edad

6.2.7 Informe de frecuencia de nuevas amistades

Aunque no formaba parte de las métricas definidas en el apartado 4.1, se incluyó este informe con el fin de conocer qué usuarios establecían con más frecuencia amistades o relaciones de coautoría en Flickr y en DBLP, respectivamente. En la Ilustración 6.12 se muestra el informe de esta métrica correspondiente a DBLP, en el que, por la información proporcionada inicialmente, tan sólo se puede conocer dicha frecuencia por año.

| DBLP - Frecuencia de coautorías |                                |                        |                             |
|---------------------------------|--------------------------------|------------------------|-----------------------------|
| Usuario                         | Nombre                         | Número de coautorías : | Nuevas coautorías por año : |
|                                 |                                | 78891                  | 0.1052                      |
| 1                               | Roberto Ciotti                 | 8                      | 0.1067                      |
| 2                               | Munetaka Saitoh                | 2                      | 0.0267                      |
| 3                               | José Antonio Portilla-Figueras | 21                     | 0.2800                      |
| 4                               | Tony S. Pollock                | 7                      | 0.0933                      |
| 5                               | Roberto Livi                   | 4                      | 0.0533                      |
| 6                               | Michael Blackstock             | 40                     | 0.5333                      |
| 7                               | Philip E. London               | 5                      | 0.0667                      |
| 8                               | Purvesh Sharma                 | 4                      | 0.0533                      |
| 9                               | Kisa Ohyama                    | 2                      | 0.0267                      |
| 10                              | R. Kantapanit                  | 4                      | 0.0533                      |
| 11                              | Kevin K. F. Yuen               | 2                      | 0.0267                      |
| 12                              | Huanmin Xu                     | 4                      | 0.0533                      |
| 13                              | Takashi Hidai                  | 3                      | 0.0400                      |
| 14                              | Dongsook Kim                   | 2                      | 0.0267                      |
| 15                              | Janet F. Johns                 | 2                      | 0.0267                      |

Ilustración 6.12 Informe de frecuencia de amistades

6.2.8 Informe de proximidad

Este informe tiene como objetivo mostrar qué porcentaje de las relaciones de amistad establecidas en cada red social se encuentran a determinadas proximidades. Además de visualizarlo en forma de tabla, como se muestra en la Ilustración 6.13, puede hacerse mediante una gráfica, como la mostrada en la Ilustración 6.14, procediendo ambos informes de Flickr. En este caso, también se permite introducir fechas como parámetro de forma que puedan compararse dichos valores entre fechas.



Ilustración 6.13 Informe de proximidad en amistades

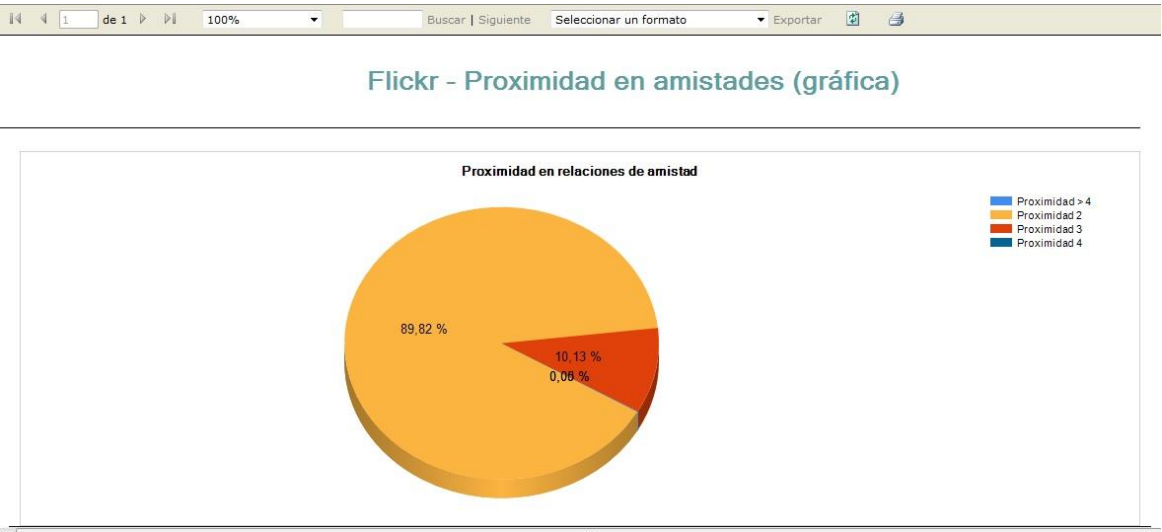


Ilustración 6.14 Gráfica de proximidad en amistades

También se incluye un informe para conocer dicho valor en las valoraciones de las fotografías, el cual puede verse en la Ilustración 6.15.



Ilustración 6.15 Informe de proximidad en votos

6.2.9 Informe de crecimiento de seguidores

El objetivo de este informe es el de mostrar el crecimiento de los amigos o seguidores de cada usuario en un periodo de tiempo determinado por el usuario de la interfaz. Implementado para Flickr y para DBLP, en el primer caso (mostrado en la Ilustración 6.16) se pueden elegir además los días de inicio y fin del periodo y ordenarlos según el crecimiento de seguidores.



Fecha de inicio: 02/11/2006 Fecha de fin: 03/12/2006

1 de 2 100% Buscar | Siguiente Seleccionar un formato

### Flickr - Crecimiento de seguidores

| Usuario | Crecimiento de seguidores |
|---------|---------------------------|
| 60      | 0,2                       |
| 36      | 0,08139535                |
| 39      | 0,07792208                |
| 74      | 0,05714286                |
| 65      | 0,05673759                |
| 32      | 0,05357143                |
| 88      | 0,04469274                |
| 66      | 0,04347826                |
| 21      | 0,04                      |
| 91      | 0,03703704                |
| 92      | 0,03647416                |
| 52      | 0,03481625                |

Ilustración 6.16 Informe de crecimiento de seguidores

#### 6.2.10 Informe del número de número de visitas, comentarios y votos recibidos por los productos de un usuario

Este informe muestra, para cada usuario, el número de votos recibidos por todos sus productos así como el número de visitas y comentarios de los mismos siempre que se disponga de ellos. En el informe mostrado en la Ilustración 6.17 y correspondiente a Flickr, se muestra junto a cada usuario el número total de votos, visitas y comentarios recibidos por sus productos pudiendo ordenar la información mostrada de mayor a menor.

Ver Propiedades Historial Suscripciones

1 de 1843 100% Buscar | Siguiente Seleccionar un formato Exportar

### Flickr - Totales por usuario

| Usuario | Número de fotografías | Número de comentarios | Número de visitas | Número de votos |
|---------|-----------------------|-----------------------|-------------------|-----------------|
| 100     | 18                    | 31                    | 5363              | 22              |
| 1000    | 71                    | 484                   | 10112             | 194             |
| 10000   | 52                    | 139                   | 6734              | 126             |
| 100001  | 12                    | 12                    | 339               | 12              |
| 100002  | 6                     | 1                     | 116               | 6               |
| 100003  | 33                    | 158                   | 8163              | 108             |
| 100004  | 1                     | 1                     | 4                 | 1               |
| 100005  | 4                     | 1                     | 77                | 4               |
| 100006  | 1                     | 0                     | 15                | 1               |
| 100008  | 26                    | 5                     | 984               | 28              |
| 100010  | 8                     | 4                     | 552               | 9               |
| 100013  | 23                    | 3                     | 1245              | 27              |
| 100017  | 2                     | 1                     | 27                | 2               |
| 100020  | 2                     | 0                     | 70                | 3               |
| 100021  | 1                     | 1                     | 7                 | 1               |
| 100023  | 9                     | 10                    | 795               | 20              |

Ilustración 6.17 Informe del total de comentarios, visitas y votos recibidos por usuario

Además, se proporciona un informe como el mostrado en la Ilustración 6.18 para conocer el máximo valor de estos datos por cada usuario.

| Ver   Propiedades   Historial   Suscripciones                     |                       |                          |                      |                    |
|-------------------------------------------------------------------|-----------------------|--------------------------|----------------------|--------------------|
| 1 de 1843 100% Buscar   Siguiente Seleccionar un formato Exportar |                       |                          |                      |                    |
| Flickr - Máximos por usuario                                      |                       |                          |                      |                    |
| Usuario                                                           | Número de fotografías | Nº máximo de comentarios | Nº máximo de visitas | Nº máximo de votos |
| 100                                                               | 18                    | 4                        | 436                  | 2                  |
| 1000                                                              | 71                    | 27                       | 569                  | 23                 |
| 10000                                                             | 52                    | 15                       | 618                  | 12                 |
| 100001                                                            | 12                    | 7                        | 62                   | 1                  |
| 100002                                                            | 6                     | 1                        | 55                   | 1                  |
| 100003                                                            | 33                    | 22                       | 641                  | 15                 |
| 100004                                                            | 1                     | 1                        | 4                    | 1                  |
| 100005                                                            | 4                     | 1                        | 26                   | 1                  |
| 100006                                                            | 1                     | 0                        | 15                   | 1                  |
| 100008                                                            | 26                    | 2                        | 302                  | 2                  |
| 100010                                                            | 8                     | 3                        | 140                  | 2                  |
| 100013                                                            | 23                    | 1                        | 310                  | 2                  |
| 100017                                                            | 2                     | 1                        | 17                   | 1                  |
| 100020                                                            | 2                     | 0                        | 39                   | 2                  |
| 100021                                                            | 1                     | 1                        | 7                    | 1                  |
| 100023                                                            | 9                     | 4                        | 298                  | 8                  |

Ilustración 6.18 Informe del máximo de comentarios, visitas y votos recibidos por usuario

### 6.2.11 Informe del número de visitas, comentarios y votos recibidos por un producto

Aunque no se definió en las métricas iniciales descritas en el apartado 4.1, se implementó este informe para MovieLens, Flickr y DBLP con el objetivo de recoger el número de votos recibidos por película en el caso de MovieLens, por publicación en DBLP y por fotografía en Flickr. En el caso de MovieLens, la información proporcionada inicialmente permite agrupar las películas por género y, a su vez, mostrar los votos agrupados por la edad, el sexo o la ocupación de los usuarios así como la valoración media otorgada, tal como se observa en la Ilustración 6.19, en la que la información de voto aparece agrupada según el sexo de los usuarios. Mostrar la información agrupada de esta forma permitirá, además, conocer patrones de preferencias entre diferentes perfiles de usuario.

|                                                                |                             |
|----------------------------------------------------------------|-----------------------------|
| Agrupar por: Sexo                                              | Fecha de inicio: 10/09/1997 |
| Fecha de fin: 23/04/1998                                       |                             |
| 1 de 2 100% Buscar   Siguiente Seleccionar un formato Exportar |                             |

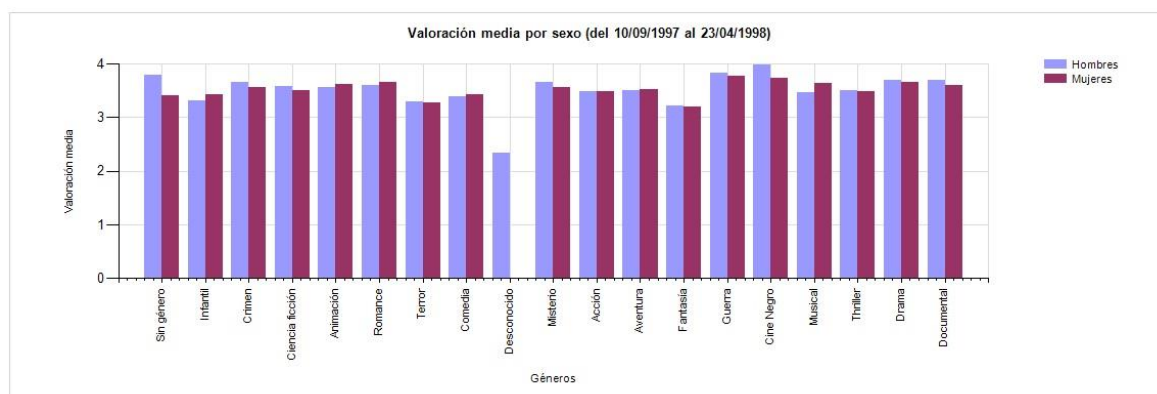
### MovieLens - Número de votos y valoración media

| Género                                              | Sexo    | Número de votos | Valoración media |
|-----------------------------------------------------|---------|-----------------|------------------|
| <input checked="" type="checkbox"/> Sin género      |         | 581             | 3.60             |
| <input checked="" type="checkbox"/> Infantil        |         | 7181            | 3.37             |
| <input type="checkbox"/> Crimen                     |         | 8034            | 3.61             |
|                                                     | Mujeres | 1787            | 3.56             |
|                                                     | Hombres | 6247            | 3.65             |
| <input checked="" type="checkbox"/> Ciencia ficción |         | 12728           | 3.54             |
| <input checked="" type="checkbox"/> Animación       |         | 3605            | 3.59             |
| <input checked="" type="checkbox"/> Romance         |         | 19454           | 3.63             |
| <input checked="" type="checkbox"/> Terror          |         | 5308            | 3.28             |

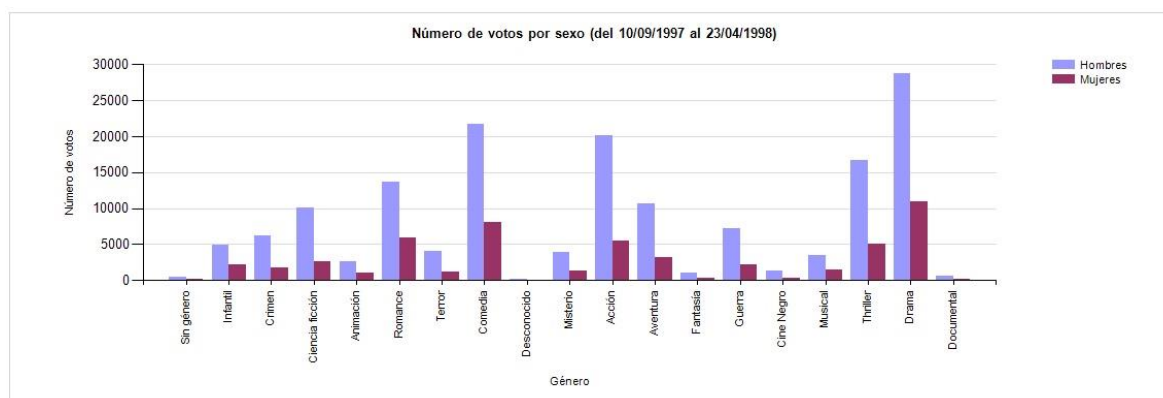
Ver gráfica

Ilustración 6.19 Informe del número de votos por producto

Esta información puede visualizarse también en forma de gráfica, tanto para la valoración media (véase Ilustración 6.20) como para el número de votos (véase Ilustración 6.21).



**Ilustración 6.20** Gráfica de la valoración media agrupada por sexo



**Ilustración 6.21** Gráfica del número de votos agrupado por sexo

### 6.2.12 Informe de reciprocidad

Este informe, implementado sólo para el conjunto de datos de Flickr y mostrado en la Ilustración 6.22, refleja el número de amistades recíprocas existentes en el periodo elegido. Además del valor por usuario, se muestra también el valor total de amistades recíprocas y el porcentaje medio de éstas a nivel de red social y se ofrece la posibilidad de ordenar a los usuarios por el número o porcentaje de amistades recíprocas.

### 6.2.13 Informe de fortaleza de los enlaces

Este informe, implementado para DBLP, muestra el valor que toma esta métrica de compromiso para un autor elegido por el usuario. A la hora de elegir dicho autor, el usuario puede hacerlo a través de un listado y, en caso de que el autor no se encuentre en dicho listado, puede introducir su nombre manualmente. El informe final, del que se puede ver un ejemplo en la Ilustración 6.23, muestra todos los autores con los que el autor analizado ha realizado publicaciones acompañados del valor que toma esta métrica con cada uno de ellos así como la media de esta medida para el autor estudiado.

Fecha de inicio: 02/11/2006 Fecha de fin: 03/12/2006

1 de 1639 100% Buscar | Siguiente Seleccionar un formato Exportar

### Flickr - Reciprocidad de amistades

| Usuario | Número de amistades reciprocas | Porcentaje |
|---------|--------------------------------|------------|
|         | 2344973                        | 25.5561 %  |
| 124457  | 21                             | 5.0000 %   |
| 121049  | 1                              | 100.0000 % |
| 35932   | 169                            | 0.5435 %   |
| 41760   | 85                             | 0.5650 %   |
| 49086   | 112                            | 0.8772 %   |
| 47389   | 34                             | 2.0833 %   |
| 78334   | 20                             | 5.8824 %   |
| 109955  | 24                             | 12.5000 %  |
| 11212   | 130                            | 0.4608 %   |
| 24073   | 10                             | 10.0000 %  |
| 269052  | 8                              | 20.0000 %  |

**Ilustración 6.22** Informe de reciprocidad de amistades

Año de inicio: 1936 Año de fin: 2010

Nombre del autor a estudiar: Roberto Ciotti Si el nombre no está en la lista, escríbalo: NULL

1 de 1 100% Buscar | Siguiente Seleccionar un formato Exportar

### DBLP - Fortaleza de las coautorías

| Coautorías                                | Ratio de publicaciones en común respecto al total de cada autor |
|-------------------------------------------|-----------------------------------------------------------------|
|                                           | 0.1160                                                          |
| Alan A. Bertossi y Roberto Ciotti         | 0.0462                                                          |
| Roberto Ciotti y Maria Cristina Pinotti   | 0.0526                                                          |
| Francesco Betti Sorbelli y Roberto Ciotti | 0.5000                                                          |
| Ferruccio Barsi y Roberto Ciotti          | 0.1250                                                          |
| Roberto Ciotti y Stephan Olariu           | 0.0091                                                          |
| Roberto Ciotti y Vlady Ravelomanana       | 0.0465                                                          |
| Roberto Ciotti y Alfredo Navarra          | 0.0323                                                          |

**Ilustración 6.23** Informe de fortaleza de los enlaces

#### 6.2.14 Informe de fidelidad de amigos

Este informe, en el que el usuario de la interfaz puede elegir las fechas de inicio y fin y el usuario a analizar, muestra la fidelidad de cada uno de los amigos de dicho usuario y el valor medio de dichas fidelidades. Se puede, además, ordenar los resultados mostrados de mayor a menor fidelidad, tal y como se muestra en la Ilustración 6.24 correspondiente al conjunto de datos Flickr.

Fecha de inicio: 02/11/2006  Fecha de fin: 03/12/2006

Usuario: 6

  1 de 3   100%   Buscar | Siguiente Seleccionar

Flickr - Fidelidad de amigos

| Usuario | Seguidor | Fidelidad ▾ |
|---------|----------|-------------|
|         |          | 0,011705    |
| 6       | 311      | 0,6667      |
| 6       | 126      | 0,1429      |
| 6       | 200      | 0,0588      |
| 6       | 261      | 0,0556      |
| 6       | 39       | 0,0476      |
| 6       | 100      | 0,0000      |
| 6       | 108      | 0,0000      |
| 6       | 10862    | 0,0000      |
| 6       | 111      | 0,0000      |
| 6       | 114      | 0,0000      |
| 6       | 118      | 0,0000      |

Ilustración 6.24 Informe de fidelidad de amigos

6.2.15 Informe del número de votos por proximidad

Esta métrica de influencia se implementa mediante informes que muestran, para cada usuario, qué número de votos recibido por sus productos procede de usuarios con diferentes proximidades así como la proximidad máxima (es decir, conocer cómo de lejos llegan las publicaciones de un usuario). Implementado para Flickr (cuyo informe se muestra en Ilustración 6.25) y DBLP permite, además, analizar los datos en diferentes periodos de tiempo.

Fecha de inicio: 02/11/2006  Fecha de fin: 03/12/2006 

  1 de 1090   100%   Buscar | Siguiente Seleccionar

Flickr - Proximidad de voto

| Usuario  | Proximidad  | Porcentaje |
|----------|-------------|------------|
| ☐ 100    | Máximo: 2   |            |
|          | 1           | 66.67 %    |
|          | 2           | 33.33 %    |
| ☐ 1000   | Máximo: 2   |            |
| ☐ 10000  | Máximo: 3   |            |
| ☐ 100001 | Máximo: 1   |            |
| ☐ 100002 | Máximo: > 6 |            |
|          | > 6         | 100.00 %   |
| ☐ 100003 | Máximo: 3   |            |
| ☐ 100008 | Máximo: 3   |            |
| ☐ 100010 | Máximo: 1   |            |
| ☐ 100013 | Máximo: 3   |            |
| ☐ 100020 | Máximo: 3   |            |
| ☐ 100023 | Máximo: 3   |            |

Ilustración 6.25 Informe de proximidad de voto

6.2.16 Listas TOP10

Puesto que una gran mayoría de sitios web ofrecen este tipo de listas, también se han incluido informes que reflejen esta información. Algunos de estos informes recogen los productos más votados (véase la Ilustración 6.26, correspondiente a las películas más votadas de MovieLens), los usuarios con más seguidores (como la Ilustración 6.27, que recoge los usuarios de Flickr con más seguidores), los usuarios con más productos publicados (en el caso de DBLP se muestra en el informe de la Ilustración 6.28), los días en los que se votan más productos (mostrado, para MovieLens, en la Ilustración 6.29) o las horas a las que se publican más productos (en el caso de MovieLens, este informe aparece en la Ilustración 6.30).

Fecha de inicio: 10/09/1997

Fecha de fin: 23/04/1998

Ver informe

Ver: Películas más y mejor valoradas

1 de 1

100%

Buscar | Siguiente

Seleccionar un formato

Exportar

MovieLens - Listas TOP10

| Posición | Película                      | Número de votos |
|----------|-------------------------------|-----------------|
| 1        | Star Wars (1977)              | 583             |
| 2        | Contact (1997)                | 509             |
| 3        | Fargo (1996)                  | 508             |
| 4        | Return of the Jedi (1983)     | 507             |
| 5        | Liar Liar (1997)              | 485             |
| 6        | English Patient, The (1996)   | 481             |
| 7        | Scream (1996)                 | 478             |
| 8        | Toy Story (1995)              | 452             |
| 9        | Air Force One (1997)          | 431             |
| 10       | Independence Day (ID4) (1996) | 429             |

| Posición | Película                                          | Valoración media | Número de votos |
|----------|---------------------------------------------------|------------------|-----------------|
| 1        | Entertaining Angels: The Dorothy Day Story (1996) | 5                | 1               |
| 2        | Great Day in Harlem, A (1994)                     | 5                | 1               |
| 3        | Marlene Dietrich: Shadow and Light (1996)         | 5                | 1               |
| 4        | Prefontaine (1997)                                | 5                | 3               |
| 5        | Saint of Fort Washington, The (1993)              | 5                | 2               |
| 6        | Santa with Muscles (1996)                         | 5                | 2               |
| 7        | Someone Else's America (1995)                     | 5                | 1               |
| 8        | Aiqing wansui (1994)                              | 5                | 1               |

Ilustración 6.26 Informe de los 10 productos más y mejor valorados

Fecha de inicio: 02/11/2006

Fecha de fin: 03/12/2006

Mostrar usuarios con más: Seguidores

1 de 1

100%

Buscar | Siguiente

Seleccionar un formato

Flickr - Listas TOP10

|    | Usuario | Número de seguidores |
|----|---------|----------------------|
| 1  | 697     | 6176                 |
| 2  | 526     | 3800                 |
| 3  | 1635    | 3533                 |
| 4  | 736     | 3368                 |
| 5  | 919     | 3116                 |
| 6  | 969     | 3104                 |
| 7  | 1242    | 3029                 |
| 8  | 3041    | 2993                 |
| 9  | 259     | 2860                 |
| 10 | 4132    | 2705                 |

Ilustración 6.27 Informe de los 10 usuarios con más seguidores

Año de inicio:

1936

Año de fin:

2011

Ver autores con más:

Publicaciones

1

de 1

100%

Buscar | Siguiente

Seleccionar un formato

Exp

DBLP - Listas TOP 10

| Posición | Nombre           | Número de publicaciones |
|----------|------------------|-------------------------|
| 1        | Sin autor        | 13581                   |
| 2        | Wei Wang         | 815                     |
| 3        | Philip S. Yu     | 594                     |
| 4        | Chin-Chen Chang  | 568                     |
| 5        | Elisa Bertino    | 565                     |
| 6        | Thomas S. Huang  | 547                     |
| 7        | Wen Gao          | 510                     |
| 8        | Wei Li           | 506                     |
| 9        | Lei Zhang        | 498                     |
| 10       | Edwin R. Hancock | 496                     |

Ilustración 6.28 Informe de los 10 usuarios con más productos

Fecha de inicio:

10/09/1997

Fecha de fin:

23/04/1998

Ver int

Ver:

Días con mayor actividad

1

de 1

100%

Buscar | Siguiente

Seleccionar un formato

Exportar

MovieLens - Listas TOP10

| Posición | Fecha      | Número de votos | Día de la semana | Periodo vacacional | Día laborable | Temporada de ventas |
|----------|------------|-----------------|------------------|--------------------|---------------|---------------------|
| 1        | 13/11/1997 | 3521            | Jueves           | No                 | Sí            | No                  |
| 2        | 01/04/1998 | 2462            | Miércoles        | No                 | Sí            | No                  |
| 3        | 14/11/1997 | 2304            | Viernes          | No                 | Sí            | No                  |
| 4        | 27/03/1998 | 1905            | Viernes          | No                 | Sí            | No                  |
| 5        | 17/12/1997 | 1758            | Miércoles        | No                 | Sí            | Navidad             |
| 6        | 31/03/1998 | 1691            | Martes           | No                 | Sí            | No                  |
| 7        | 30/03/1998 | 1605            | Lunes            | No                 | Sí            | No                  |
| 8        | 21/11/1997 | 1553            | Viernes          | No                 | Sí            | No                  |
| 9        | 07/01/1998 | 1303            | Miércoles        | No                 | Sí            | No                  |
| 10       | 02/04/1998 | 1297            | Jueves           | No                 | Sí            | No                  |

Ilustración 6.29 Informe de los 10 días con mayor actividad



Fecha de inicio: 10/09/1997 Fecha de fin: 23/04/1998 Ver informe

Ver: Horas con mayor actividad

1 de 1 100% Buscar | Siguiente Seleccionar un formato Exportar

### MovieLens - Listas TOP10

| Posición | Hora     | Número de votos |
|----------|----------|-----------------|
| 1        | 22:26:00 | 175             |
| 2        | 22:22:00 | 174             |
| 3        | 22:43:00 | 172             |
| 4        | 22:20:00 | 168             |
| 5        | 22:07:00 | 161             |
| 6        | 20:08:00 | 161             |
| 7        | 22:29:00 | 161             |
| 8        | 22:44:00 | 161             |
| 9        | 22:16:00 | 159             |
| 10       | 22:28:00 | 159             |

**Ilustración 6.30** Informe de las 10 horas con mayor actividad

## 6.3 Seguridad

Con el fin de evitar que usuarios que no conocen por completo las funcionalidades del servidor web o que usuarios malintencionados generen problemas al realizar consciente o inconscientemente ciertas acciones sobre los informes mostrados, Reporting Services permite la creación de diferentes usuarios o roles de forma que se pueda restringir su libertad a la hora de realizar ciertas acciones sobre la interfaz. Entre estas acciones están, por ejemplo, la creación y eliminación de informes y carpetas o la visualización de estos. Este último caso es importante si, por ejemplo, se dispusiera de información protegida de los usuarios que forman parte de los conjuntos de datos estudiados. De esta forma, se podrían restringir los accesos a los informes o carpetas que contuvieran estos datos sensibles. Aunque en los conjuntos de datos empleados en este proyecto no se ha dado este último caso, sí se ha creado un usuario con permisos limitados de forma que sus acciones en la interfaz se limiten a visualizar los informes creados. En la Ilustración 6.31 se muestra cómo ve la interfaz este usuario mientras que la Ilustración 6.32 corresponde a un usuario con permisos de administrador. Como se observa en esta última ilustración, el administrador tiene la posibilidad de realizar ciertas acciones, como la creación de informes, de carpetas o de orígenes de datos, que no están disponibles en la interfaz del usuario.

Métricas para el análisis de redes sociales

Inicio | Mis suscripciones | Ayuda

Buscar:

Contenido

Mostrar detalles

**DBLP**

Informes realizados sobre el conjunto de datos DBLP, que almacena información referente a artículos relacionados con la computación

**Flickr**

Informes realizados sobre la red social Flickr, que almacena información referente a esta red social centrada en la publicación de fotografías.

**Métricas de alcance**

Informes basados en métricas cuya finalidad es la de conocer el grado de propagación efectiva de cierto contenido (por ejemplo, crecimiento de usuarios, velocidad de propagación o frecuencia de voto).

**Métricas de compromiso**

Informes basados en métricas cuya finalidad es conocer el grado de participación e implicación de los usuarios (por ejemplo, el número de visitas, comentarios y votos que reciben los productos de un usuarios o su número de seguidores)

**Métricas de influencia**

Informes basados en métricas cuya finalidad es conocer qué influencia o grado de movilización puede causar un usuario sobre otros (por ejemplo, cuántos nuevos usuarios se dan de alta tras una invitación o hasta dónde llegan los productos de un usuario)

**MovieLens**

Informes realizados sobre la red social MovieLens, que recoge información sobre la valoración que sus usuarios realizan sobre películas

**Ilustración 6.31** Interfaz web vista por un usuario



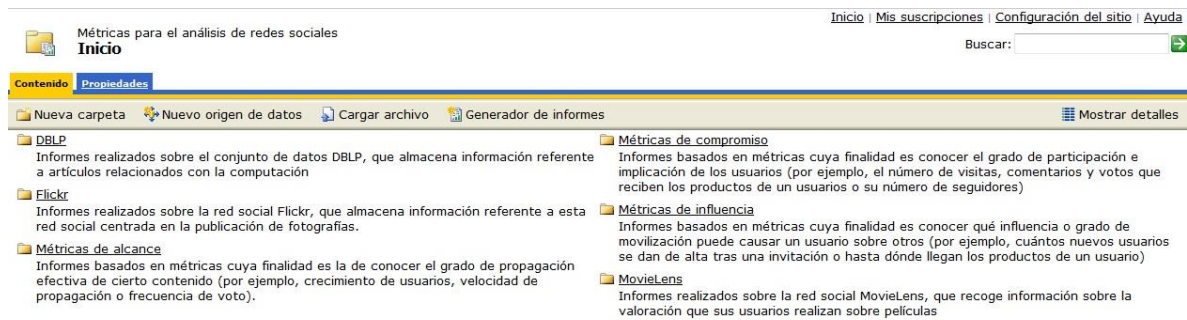


Ilustración 6.32 Interfaz web vista por un administrador

## 6.4 Mejoras de rendimiento

Debido al volumen del almacén de datos la generación de ciertos informes consumía mucho tiempo y recursos, por lo que fue necesaria la implementación de ciertas acciones que contribuyeran a mejorar el rendimiento. Una de estas acciones fue la creación de cubos OLAP que ayudasen en la generación de algunos de estos informes. En la Ilustración 6.33 se muestra el cubo diseñado, mediante Analysis Services, para los informes encargados de mostrar el número de votos, visitas y comentarios recibido por los productos de cada usuario y que se describieron en los apartados 6.2.10 y 6.2.11.

Además del uso de un cubo OLAP, se crearon índices sobre las columnas del almacén de datos que más intervenían en las consultas con el fin de agilizar éstas. Por ejemplo, fueron necesarios índices en aquellos atributos utilizados para diferenciar la procedencia de los datos debido a su alta presencia en las consultas, como las columnas *Name* y *Gender* de la tabla *USER\_DIMENSION* y la columna *UserKey* de *PRODUCT\_DIMENSION*, así como en aquellas columnas relacionadas con parámetros habituales, como es el caso de *DateColumn* en *DATE\_DIMENSION*.

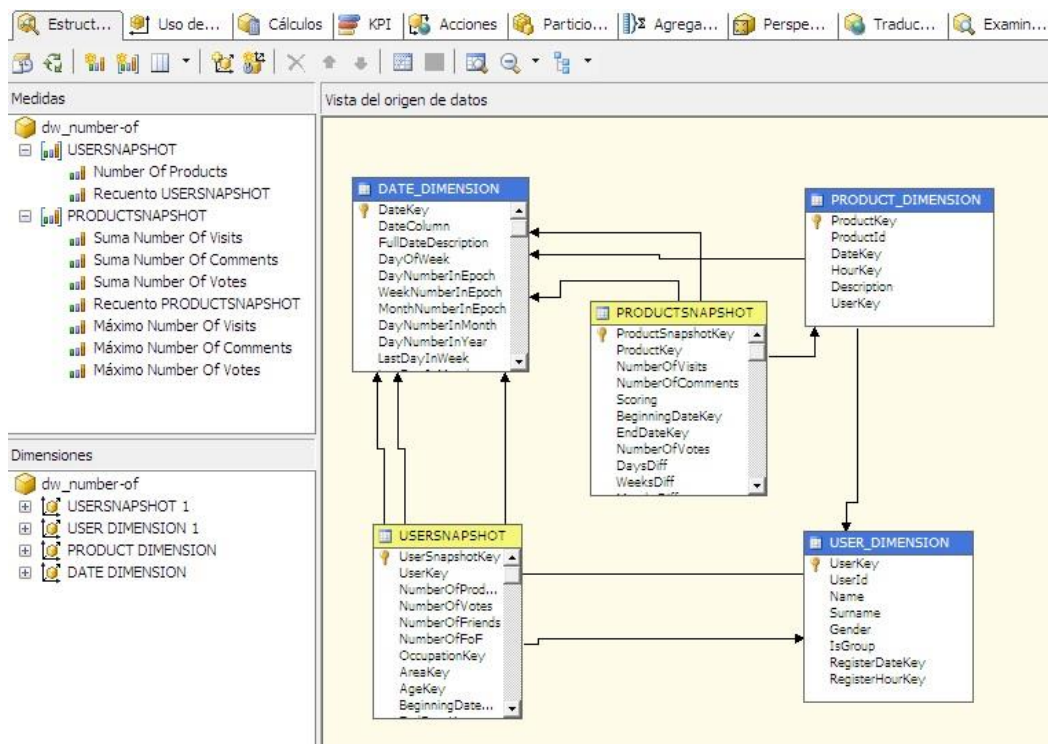


Ilustración 6.33 Cubo diseñado para calcular el número de votos recibido por usuario

## 7 Conclusiones y líneas futuras

En este capítulo se resumen las conclusiones a las que se ha llegado tras la realización del proyecto así como una serie de posibles mejoras a realizar en el futuro sobre el trabajo realizado.

### 7.1 Conclusiones

Este proyecto tenía como objetivo principal la implementación de una solución Business Intelligence suficientemente genérica que calculara y mostrara una serie de métricas útiles para el análisis de redes sociales y orientadas hacia la toma de decisiones. Para ello, se trabajó con tres conjuntos de datos procedentes de redes sociales con muy distinto fin, se diseñó un almacén de datos que recogiera la información necesaria para el cálculo de los indicadores, se programó este cálculo y se diseñaron informes web para mostrar estos indicadores.

A la hora de almacenar estos conjuntos de datos se utilizó, en primer lugar, un gestor de bases de datos orientadas a grafos, Neo4J. El uso de este gestor permitió, por un lado, almacenar los datos en una estructura utilizada generalmente por las redes sociales y, por otro, la obtención de medidas cuyo cálculo habría resultado muy complejo o incluso imposible si no se hubiera hecho uso de este tipo de bases de datos. Por estar implementado en Java, se pudo hacer uso de clases y métodos de este lenguaje de programación, lo que simplificó ciertas tareas como, por ejemplo, el almacenamiento de los datos extraídos en ficheros.

Si bien el uso de Neo4J facilitó el cálculo de ciertas medidas, los conjuntos de datos elegidos eran de gran tamaño, lo cual perjudicó al rendimiento ya que, a pesar de que los grafos facilitan este tipo de cálculos, hubiese sido necesario el uso de una máquina con mayores prestaciones para alcanzar el rendimiento óptimo ofrecido por el gestor.

Este problema ocasionado por la gran cantidad de datos utilizados también repercutió en la tarea de carga de los datos extraídos en la base de datos relacional. Gracias al uso de medidas propias de las soluciones Business Intelligence, como la creación de *data staging*, y de la plataforma que ofrece SQL Server para el proceso ETL, Integration Services, se consiguió que el manejo de cantidades ingentes de datos no afectara sobremanera a este proceso.

Respecto a la interfaz de usuario, se optó por utilizar Reporting Services por su facilidad para conectar con bases de datos sobre las que realizar las consultas y la sencillez de su interfaz para la visualización de resultados. Sin embargo, esta sencillez limitó en gran medida la personalización de los informes. Si bien permite el uso de parámetros y la modificación de aspectos referentes al formato, la personalización de los informes por parte del usuario final es difícil o incluso imposible de implementar. Por ejemplo, si el usuario quisiera ampliar una zona concreta de un gráfico de barras, es necesario incluir parámetros referentes a la información a mostrar, lo que lleva a realizar de nuevo la consulta.

En cuanto a la generación de informes, puedo decir que a los problemas asociados con el volumen de datos se les unió la complejidad de las consultas que se debían realizar

lo que provocó que, en algunas ocasiones, el tiempo de espera para visualizarlos fuese demasiado alto. Este hecho me obligó a hacer uso de la tecnología OLAP.

Con todo ello, la herramienta desarrollada demuestra que la solución de inteligencia de negocio diseñada permite dar respuesta a las métricas a alcanzabilidad, compromiso e influencia definidas, las cuales son de interés para cualquier empresa u organismo que esté presente en las redes sociales.

## 7.2 Líneas futuras

Aunque se han cumplido los objetivos fijados al inicio del proyecto, durante su desarrollo y una vez finalizado el mismo, se han detectado ciertas mejoras en lo referente a la visualización de resultados y a ampliar los requisitos a los que la solución debe dar respuesta.

- **Mejora de la interfaz de visualización de informes.** Como se mencionó en el apartado 7.1, aunque la herramienta Reporting Services ofrece un gran número de facilidades a la hora de implementar informes, la personalización de estos es muy limitada. Por ejemplo, a la hora de mostrar los informes en la interfaz web hay que adaptarse al formato establecido por la herramienta, como por ejemplo el modo en que se muestran las carpetas o el orden de éstas. Por este motivo, pueden utilizarse otras opciones que ofrece Reporting Services a la hora de acceder y explotar los informes, las cuales implican la programación de una interfaz.
- **Tareas de mantenimiento automáticas.** Aunque con los datos empleados en este proyecto no ha sido necesario incluir estas tareas puesto que se trabaja con la información acumulada en periodos de tiempo prolongados, podría darse el caso de que fuera necesario realizar la carga de datos desde el grafo hasta el almacén a diario. Por ello, la implementación mediante Integration Services de tareas automáticas de actualización y de realización de copias de seguridad sería un aspecto fundamental a tener en cuenta.
- **Ampliación de la lista de métricas.** Aunque las métricas definidas pueden aplicarse a muchas redes sociales, el gran número, variedad y crecimiento de éstas en la actualidad favorece la ampliación de dichas métricas de forma que recojan las nuevas características que van apareciendo con la expansión de estas redes.
- **Aplicación de la solución propuesta a un caso real.** Si bien se han utilizado conjuntos de datos procedentes de redes sociales reales, los datos obtenidos de éstas no contenían toda la información necesaria y, en muchas ocasiones, existían errores en su contenido provocado, en gran medida, por tratarse de conjuntos de datos públicos. Por ello, podría ofrecerse la solución creada a organizaciones o usuarios individuales con perfiles en redes sociales que proporcionaran datos completos y correctos para extraer de ellos las métricas definidas de forma que, posteriormente, pudieran utilizar la información obtenida con dichas métricas para mejorar o cambiar características de sus perfiles.

## Bibliografía

- OLAP Council. (1993). Obtenido de <http://www.olapcouncil.org>
- Amann, B., & Scholl, M. (1992). Gram: A Graph Data Model and Query Language. *European Conference on Hypertext Technology (ECHT)* (págs. 201-211). ACM.
- Andries, M., Gemis, M., Paredaens, J., Thyssens, I., & Den Bussche, J. V. (1992). Concepts for graph-oriented object manipulation. *Proceedings of the 3rd International Conference on Extending Database Technology (EDBT)* (págs. 21-38). Springer.
- Angles, R., & Gutiérrez, C. (Febrero 2008). Survey of Graph Database Models. *ACM Computer Surveys*.
- Awareness Inc. (2012). Actionable Social Analytics: From Social Media Metrics to Business Insights.
- Cano Giner, J. L. (2007). *Business Intelligence: Competir con información*. Madrid: Fundación Cultural Banesto.
- Codd, E. F. (1970). A relational model of data for large shared data banks. *Commun. ACM* 13, 377-387.
- Codd, E. F. (1980). Data models in database management. *Proceedings of the 1980 Workshop on Data abstraction, Databases, and Conceptual Modeling*, 112-114.
- Fuchs, G. (2003). *Aspects of ROI*.
- Gartner Group. (Enero de 2006). *Glosario*. Obtenido de <http://www.gartnergroup.com>
- Gemis, M., & Paredaens, J. (1993). An object-oriented pattern matching language. *Proceedings of the First JSSST International Symposium on Object Technologies for Advanced Software* (págs. 339-355). Springer-Verlag.
- Graves, M., Bergeman, E. R., & Lawrence, C. B. (1995). A graph-theoretic data model for genome mapping databases. *Proceedings of the 28th Hawaii International Conference on System Sciences (HICSS)*.
- Gruhl, D., Nagarajan, M., Pieper, J., Robson, C., & Sheth, A. (2010). Multimodal Social Intelligence in a Real-Time Dashboard System. *The VLDB Journal — The International Journal on Very Large Data Bases*, 825-848.
- Gutiérrez, A., Pucheral, P. S., Thévenin, & J.M. (1994). Database graph views: A practical model to manage persistent graphs. *Proceedings of the 20th International Conference on Very Large Data Bases (VLDB)* (págs. 391-402). Morgan Kaufmann.
- Güting, R. H. (1994). GraphDB: modeling and querying graphs in databases. *Proceedings of the 20th International Conference on Very Large Data Bases (VLDB)* (págs. 297-308). Morgan Kaufmann.

- Gyssens, M., Paredaens, J., Den Bussche, J. V., & Gucht, D. V. (1990). A graph-oriented object database model. *Proceedings of the 9th Symposium on Principles of Database Systems (PODS)* (págs. 417-424). ACM Press.
- Hidders, J. (2002). Typing graph-manipulation operations. *Proceedings of the 9th International Conference on Database Theory (ICDT)* (págs. 394-409). Springer-Verlag.
- Hidders, J., & Paredaens, J. (1993). GOAL, A graph-based object and association language. *Advances in Database Systems: Implementations and Applications*, 247-265.
- Hill, S., Provost, F., & Volinsky, C. (2006). Network-Based Marketing: Identifying Likely Adopters via Consumer Networks. *Statistical Science*, 256-276.
- Inmon, B. (2005). *Building the Data Warehouse*. Wiley.
- Inmon, B. (2006). Business Intelligence Network.
- Kazienko, P., Musial, K., Kukla, E., Kajdanowicz, T., & Bródka, P. (2011). Multidimensional Social Network: Model and Analysis. *Computational Collective Intelligence. Technologies and Applications*, 378-387.
- Kimbal, R., & Ross, M. (2002). *The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling*. Wiley John & Sons.
- Knight, B., Veerman, E., Dickinson, G., Hinson, D., & Herbold, D. (2008). *Microsoft SQL Server 2008 Integration Services*. Indianapolis, Indiana: Wiley Publishing, Inc.
- Kunii, H. S. (1987). DBMS with graph data model for knowledge handling. *Proceedings of the 1987 Full Joint Computer Conference on Exploring technology: Today and Tomorrow* (págs. 138-142). IEEE Computer Society Press.
- Kuper, G. M., & Vardi, M. Y. (1984). A new approach to database logic. *Proceedings of the 3th Symposium on Principles of Database Systems (PODS)* (págs. 86-96). ACM Press.
- Larson, B. (2008). *Microsoft SQL Server 2008 Reporting Services*. Emeryville, California: McGraw-Hill/Osborne.
- Lécluse, C., Richard, P., & Vélez, F. (1988). O2, an object-oriented data model. *Proceedings of the ACM SIGMOD International Conference on Management of Data* (págs. 424-433). ACM Press.
- Levene, M., & Loizou, G. (1995). A graph-based data model and its ramifications. *IEEE Trans. Knowl. Data Eng.*, 809-823.
- Levene, M., & Poulouvasilis, A. (1990). The Hypernode model and its associated query language. *Proceedings of the 5th Jerusalem Conference on Information technology* (págs. 520-530). IEEE Computer Society Press.
- Levene, M., & Poulouvasilis, A. (1991). An object-oriented data model formalised through hypergraphs. *Data Knowl. Eng.* 6, 205-224.

- Mainguenaud, M. (1992). Simatic XT: A data model to deal with multi-scaled networks. *Comput. Environ. Urban Sys.* 16, 281-288.
- Mislove, A., Marcon, M., Gummadi, K. P., Druschel, P., & Bhattacharjee, B. (2007). Measurement and Analysis of Online Social Networks. *ACM SIGCOMM Conference on Internet Measurement* (págs. 29-42). San Diego, California: ACM.
- Nguyen, Q. V., & Huang, M. L. (2010). A New Interactive Platform for Visual Analytics of Social Networks. *Visual Information Communication*, 231-244.
- Palazuelos, C., & Zorrilla, M. (2011). FRINGE: A New Approach to the Detection of Overlapping Communities in Graphs. *ICCSA'11 Proceedings of the 2011 international conference on Computational science and its applications - Volume Part III*, 638-653.
- Palazuelos, C., & Zorrilla, M. (2012). Analysis of Social Metrics in Dynamic Networks: Measuring the Influence with FRINGE. *EDBT-ICDT '12 Proceedings of the 2012 Joint EDBT/ICDT Workshops*, 9-12.
- Papakonstantinou, Y., García-Molina, H., & Andwidom, J. (1995). Object exchange across heterogeneous information sources. *Proceedings of the 11th International Conference on Data Engineering (ICDE)*. (págs. 251-260). IEEE Computer Society.
- Paredaens, J., Peelman, P., & Tanca, L. (1995). G-Log: A graph-based query language. *IEEE Trans. Knowl. Data Eng.* 7, 436-453.
- Roussopoulos, N., & Mylopoulos, J. (1975). Using semantic networks for database management. *Proceedings of the International Conference on Very Large Databases (VLDB)* (págs. 144-172). ACM.
- The Data Warehouse Institute. (s.f.). *The Data Warehouse Institute*. Obtenido de <http://tdwi.org/Home.aspx>
- Ting, I.-H., Lin, C.-H., & Wang, C.-S. (2011). Constructing A Cloud Computing Based Social Networks Data Warehousing and Analyzing System. *International Conference on Advances in Social Networks Analysis and Mining* (págs. 735-740). IEEE Computer Society.
- Watson, H. J. (Agosto de 2006). *Terry College Of Business*. Obtenido de [http://www.terry.uga.edu/~pmckeown/RevMan05/Data\\_warehousing\\_Revenue\\_Management.ppt](http://www.terry.uga.edu/~pmckeown/RevMan05/Data_warehousing_Revenue_Management.ppt)
- White, J. S., Matthews, J. M., & Stacy, J. L. (2012). Coalmine: An Experience in Building a System for Social Media. *Proc. SPIE 8408, Cyber Sensing 2012*.
- Whittemore, B. (2003). *The Business Intelligence ROI Challenge: Putting It All Together*.
- Zuckerberg, M. (2007). Social Graph. *Facebook F8*.

## Anexo I. Captura del almacén de datos diseñado

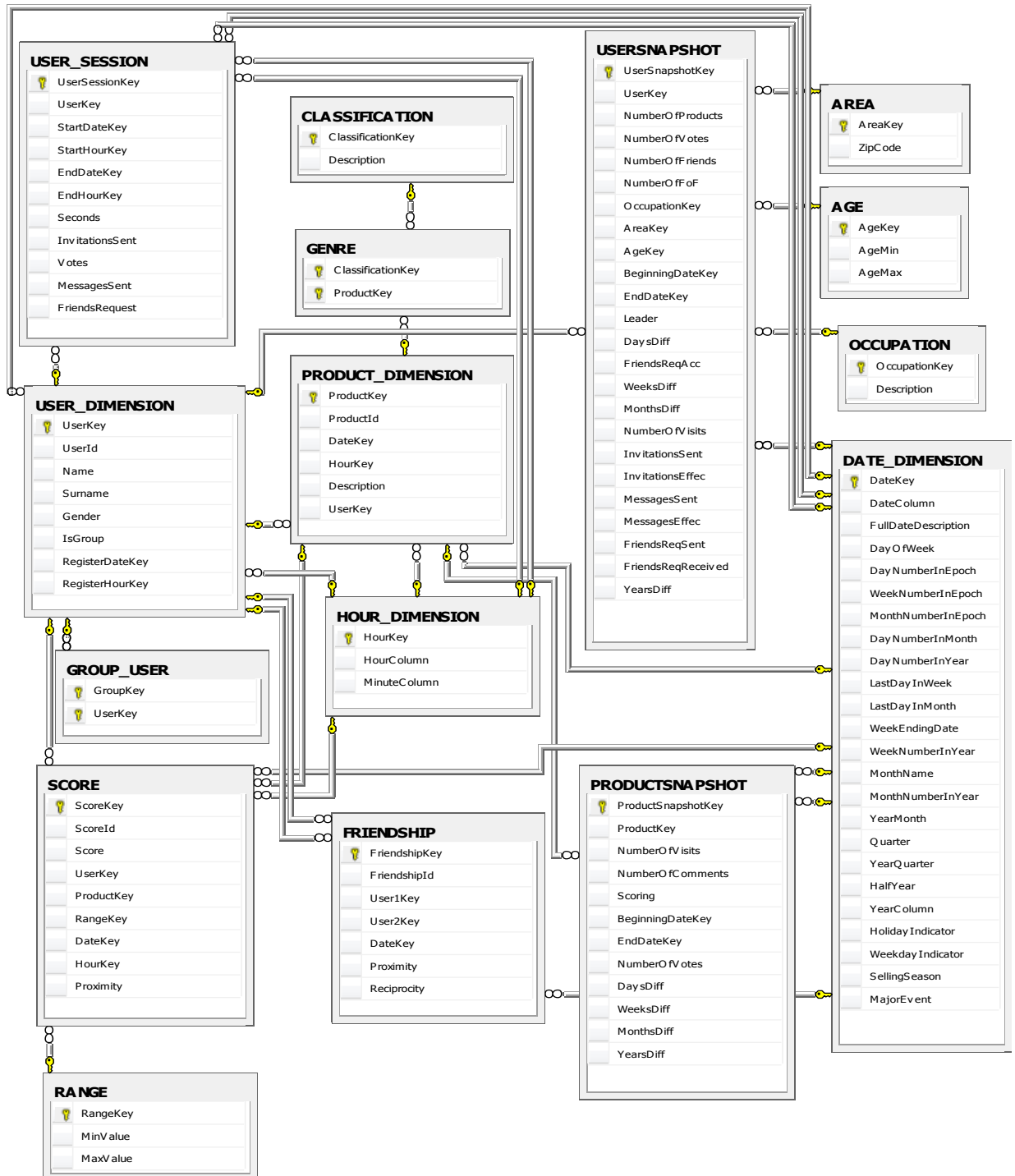


Ilustración A.1 Diseño del almacén de datos